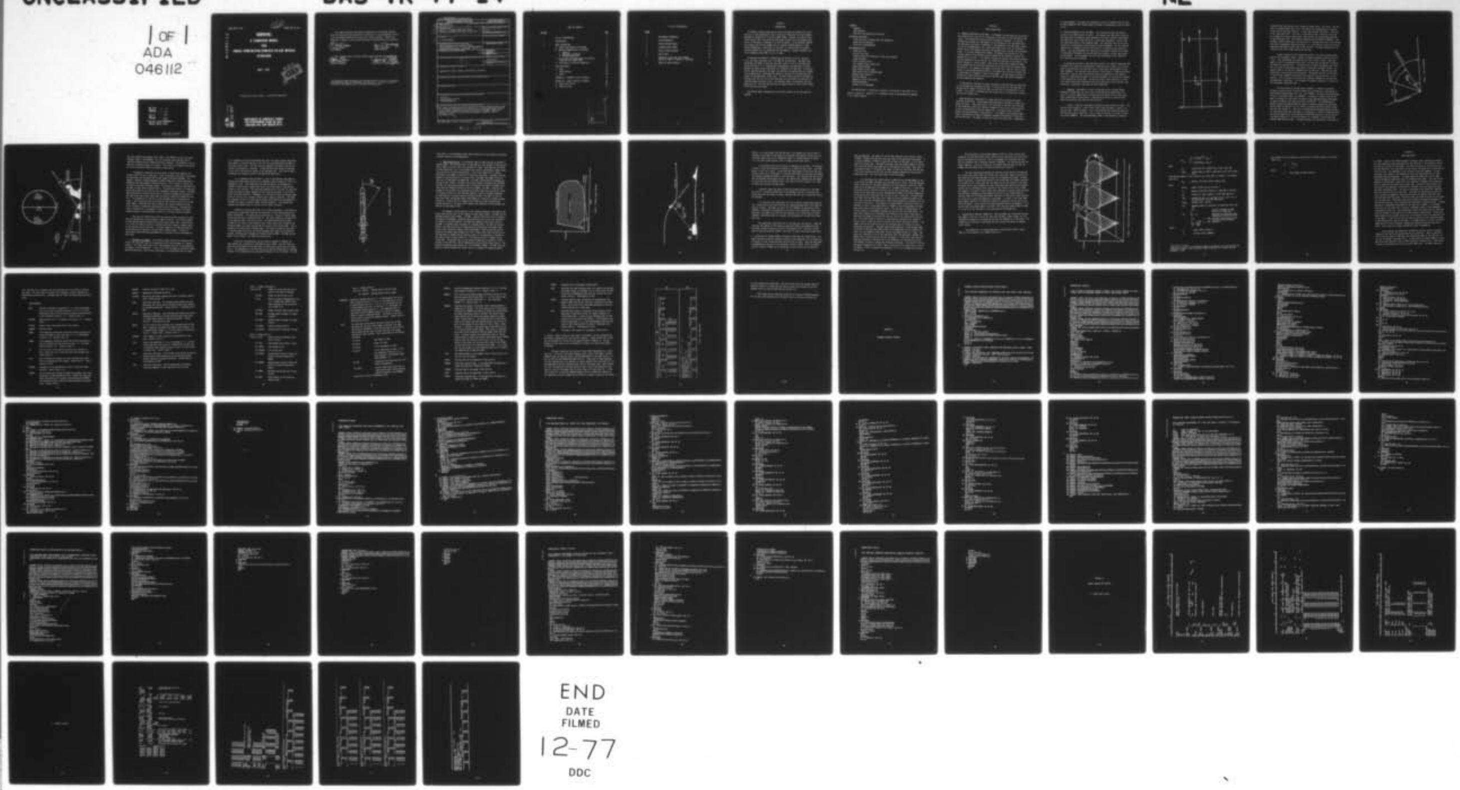
AD-A046 112 DIRECTORATE OF AEROSPACE STUDIES KIRTLAND AFB N MEX F/G 1/3
SURVIVE: A COMPUTER MODEL FOR SINGLE PENETRATOR/SURFACE-TO-AIR --ETC(U)
JUL 77 A F COOPER

UNCLASSIFIED DAS-TR-77-14 NL

1 OF 1
ADA
046112



END
DATE
FILMED
12-77
DDC

DAS-TR-77-14 DAS-TR-77-14

SURVIVE:

A COMPUTER MODEL FOR SINGLE PENETRATOR/SURFACE-TO-AIR MISSILE ATTRITION

JULY 1977

APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

DIRECTORATE OF AEROSPACE STUDIES
DCS/DEVELOPMENT PLANS HQ AFSC
KIRTLAND AFB, NEW MEXICO 87117

This report discusses the theory and operation of the SURVIVE computer model for evaluating the probability of survival of a single penetrator flying in an environment defended by surface-to-air missile systems. SURVIVE was developed in support of the Utility Evaluation of Stand-Off Missile Candidates study.

A. Foster Cooper

A. FOSTER COOPER
Study Member

Hans J. Gevelhoff

HANS J. GEVELHOFF
Study Director

This Technical Report has been reviewed and is approved for publication.

Harry L. Gogan

HARRY L. GOGAN
Technical Director

James P. Ditz

JAMES P. DITZ, Lt Col, USAF
Director of Aerospace Studies

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 14 DAS-TR-77-14	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 SURVIVE: A Computer Model for Single Penetrator/Surface-to-Air Missile Attrition.		5. TYPE OF REPORT & PERIOD COVERED 9 Final rept., 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 A. Foster/Cooper		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFSC Directorate of Aerospace Studies DCS/Development Plans, Air Force Systems Command Kirtland Air Force Base, New Mexico 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		11 12. REPORT DATE Jul[redacted] [redacted]77 13. NUMBER OF PAGES 70
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 73p.		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Attrition
Surface-to-Air Missile
Single Penetrator

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report discusses the theory and operation of the SURVIVE computer model for evaluating the probability of survival of a single penetrator flying in an environment defended by surface-to-air missile systems. SURVIVE was developed in support of the Utility Evaluation of Stand-Off Missile Candidates study.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 237 LB

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The SURVIVE computer model may be used to evaluate the probability of survival of a single penetrator flying a specified flight path in an environment defended by surface-to-air missile systems (SAMS). Options of scenario, coordinate system, SAM firing doctrine, and target location give the program flexibility in the types of problems it is able to handle. At one extreme, SURVIVE can evaluate the survival probability of a weapon launched against a fixed target defended by a single SAM; at the other, it can evaluate the expected survival probability of a penetrator and the weapon it launches through a corridor defended by many SAMs of up to 10 types.

The defense environment in SURVIVE may be specified in one of two ways. First, the exact locations of all the SAMs can be specified. The model then determines the number of missiles each SAM is able to fire at the penetrator during the time it is in coverage and, subsequently, the penetrator survival probability based on the single shot kill probability of each SAM. In addition to this specific approach, the model can calculate an expected survival probability by generating a representative sample of locations of the SAM sites with respect to the penetrator flight path. The model then determines the number of missiles each site can fire at the penetrator and calculates an average for all locations where intercepts occur. The expected number of encounters that the penetrator will have with each SAM and the expected value of the survival probability are then calculated.

The SURVIVE model incorporates the following aspects of the survivability problem:

SCENARIO

- Geometry
- SAM placement
- Masking of the penetrator by terrain

PENETRATOR AND WEAPON

- Flight profiles (3-dimensional time dependent)
- Radar and IR signatures
- Electronic countermeasures

SAM CHARACTERISTICS

- Salvos per site
- Missiles per salvo
- Timing criteria for tracking, firing, and reloading
- Lethal envelope
- Missile flyout profile
- Kill probabilities, single shot
- Radar antenna height
- Radar ground clutter angle
- Radar sensitivity or maximum range
- Maximum radar elevation
- Missile guidance
- Geometric launch restrictions
- ECM effectiveness
- Probability of engagement

The SURVIVE model is described in section 2; the inputs to the model and its output in section 3. Appendix A is a FORTRAN listing of the program and appendix B is a sample problem.

SECTION 2

MODEL DESCRIPTION

A. GENERAL DESCRIPTION OF THE MODEL. Two methods are applicable to calculating the probability of survival of a penetrator flying over an area defended by SAM sites: the deterministic method and the probabilistic method. In a purely deterministic method, all the aspects of the problem that are considered would be simulated by the model to determine if the penetrator survived under a specific set of circumstances or not. Many cases would have to be run to obtain a statistically valid sample from which the probability of survival could be determined. In a purely probabilistic method, the probability functions for the various aspects of the problem would be determined and combined to obtain the probability of survival. This tends to obscure the effects that specific aspects of the problem have on the results, but simplifies modeling of the problem. The SURVIVE program is a hybrid model. Many of the interactions between the SAMs and the penetrator are simulated deterministically and others are handled probabilistically. For example, the number of missiles that a SAM can fire at a penetrator is calculated deterministically, but whether the site will actually fire is controlled by a probability input to the model. Two approaches are used for locating the flight profiles relative to the SAM sites: the specific approach and the expected value approach.

In the specific approach, the exact locations of the SAM sites are specified in a two-dimensional coordinate system. A flight profile for the penetrator is defined and the number of missiles each site can fire at the penetrator is calculated. This value is combined with the single-shot kill probability to obtain the survival probability.

Under battlefield conditions it is most difficult to pinpoint the exact locations of SAM sites. Intelligence concerning their locations is usually outdated and scanty. Such uncertainty in the locations of the SAM sites relative to the penetrator flight path requires a number of cases to be run and averaged to give a representative survival probability. This would prove to be both tedious and time consuming if done manually. The expected value approach generates a representative sample of geometries between the penetrator flight path and the SAM sites. The number of missiles fired by the SAM sites is determined

for each geometry. The model then computes the survival probabilities for each of these geometries and averages them to provide a representative survival probability.

B. DETAILED CAPABILITIES OF THE MODEL. This section discusses the various aspects of the SURVIVE model in detail. The program uses identifying labels on the input cards to associate input parameters with their function in the model. The identifier may be associated with a single input parameter, in which case the identifying label is the parameter name. The identifier may also be associated with a functionally related group of input parameters. In this case the identifier is a mnemonic device to aid the user in formatting the input parameters. The input parameters and identifiers discussed in section 3 are given in parentheses throughout this section when the concepts to which they relate are discussed. Input parameters are also cross-referenced to the identifier with which they are associated.

As mentioned previously, the model may be used in its specific mode when SAM placement with respect to the penetrator flight profile is known, or it can be used in the expected value mode when less is known about the placement of the SAMs (AVERAGE). The expected value approach is more useful in developing system evaluation criteria, since the resulting survival probability is an average based on a representative number of relative placements of the SAMs with respect to the penetration flight path. The specific mode is most effective in situations where operational information would give some idea of SAM locations, thus indicating a preferred flight path to avoid those SAMs.

1. Scenario. Rectangular or polar coordinates may be selected (AREA). Rectangular coordinates are most useful in describing the penetration of an area or a corridor where there is a preferred orientation of the flight path. Polar coordinates are more useful in describing an attack on a point target where the penetrator may come from any direction.

Figure 1 shows the rectangular coordinate system used by the model. The SAM sites are located in X and Y (XSITE, YSITE) when the model is used in its specific mode. When the model is run in the expected value mode the density of the SAM sites must be specified. This is done by defining a corridor of specified width (CORWDTH). The area defended by SAMs is then defined by locating

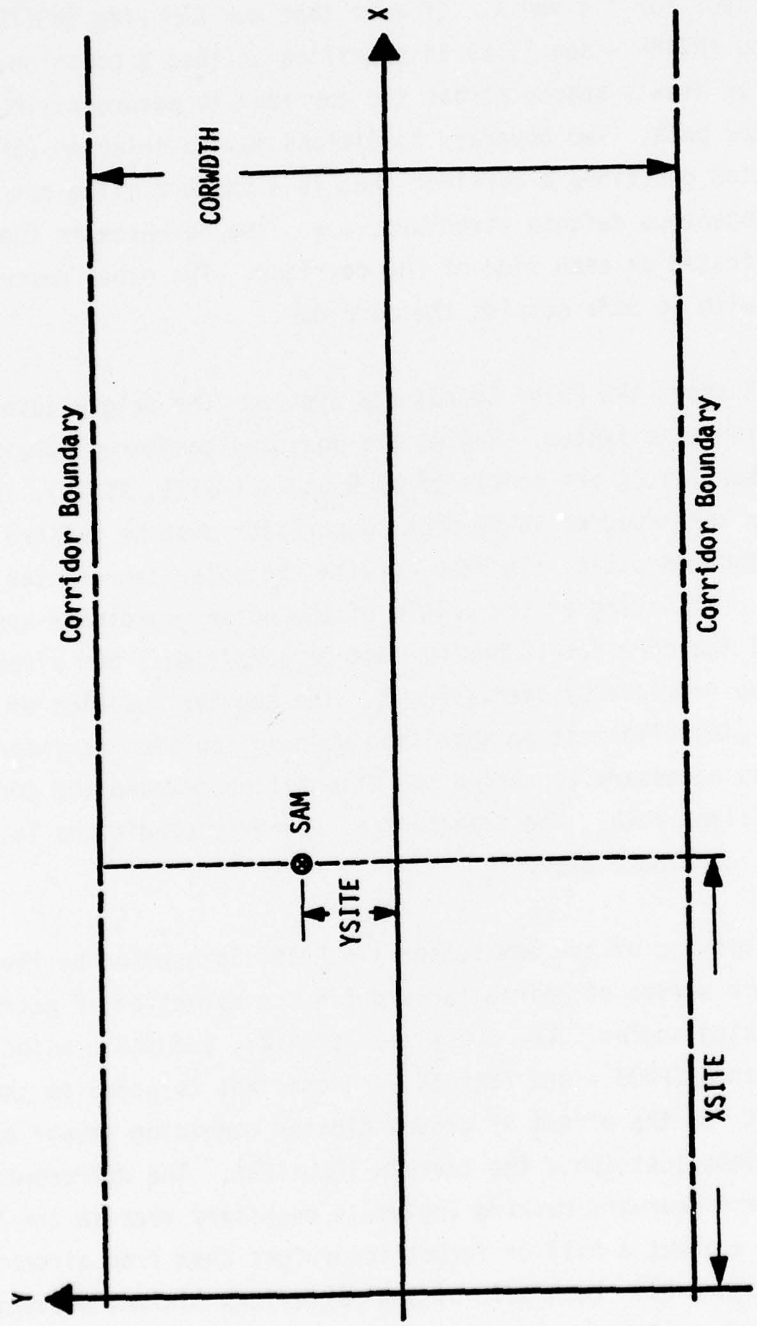


Figure 1. Rectangular Coordinates

the SAMs within the corridor as a function of depth (XSITE - see SITE). The corridor is centered about the X axis. The SAM is able to assume any Y position across the corridor for a given X. If more than one SAM site (NSITE - see SITE) of the same type (NTYPE - see SITE) is specified at that X position, the sites are assumed to be evenly spaced across the corridor to assure maximum coverage of the penetrator path. Two boundary conditions may be selected (SYMETRY). One boundary condition describes a corridor that is a uniform slice out of an infinite area with a homogeneous defense structure, i.e., the defenses in the corridor are repeatedly duplicated on each side of the corridor. The other describes an isolated corridor with no SAMs outside the corridor.

Figure 2 shows the Polar Coordinate System. The origin coincides with that of the rectangular system. Angles are measured counter-clockwise from the X axis. Site coordinates are specified by R and θ (YSITE, XSITE). When the model is used in the expected value mode, a corridor must be defined to specify the density of the SAM sites. In keeping with the polar coordinates, the corridor is a sector originating at the origin of the polar coordinate system. The angular width of the corridor (CORWDTH) must be specified. SAM sites are then positioned in the corridor by specifying R. The angular position of the center of the corridor (TARGETY) must be specified when an isolated corridor is considered. This is necessary to define the orientation between the corridor and the penetrator flight path. The treatment of boundary conditions is similar to the rectangular coordinate case.

Terrain masking of the SAM sensor (TERRAIN) is handled by the model through a discrete series of paired forward (in the direction of decreasing X) and rearward masking angles (ELF, ELR - see TERRAIN), and their associated probability of occurrence (PROB - see TERRAIN). A constant is added to the terrain angles to account for the effect of ground clutter degrading sensor performance at sensor elevations just above the terrain (CLUTTER). The differentiation between forward and rearward masking angles is necessary because the SAM sites may be backed up against a hill or forest to protect them from aircraft coming from the rear (figure 3). Each pair of angles defines minimum elevation angles for the sensor below which the sensor is ineffective. A series of masking angles and their probability of occurrence may be specified because in reality the terrain masking angles will not be the same for all azimuths in the forward and rearward masking angle areas. The masking angles are specified as a function of

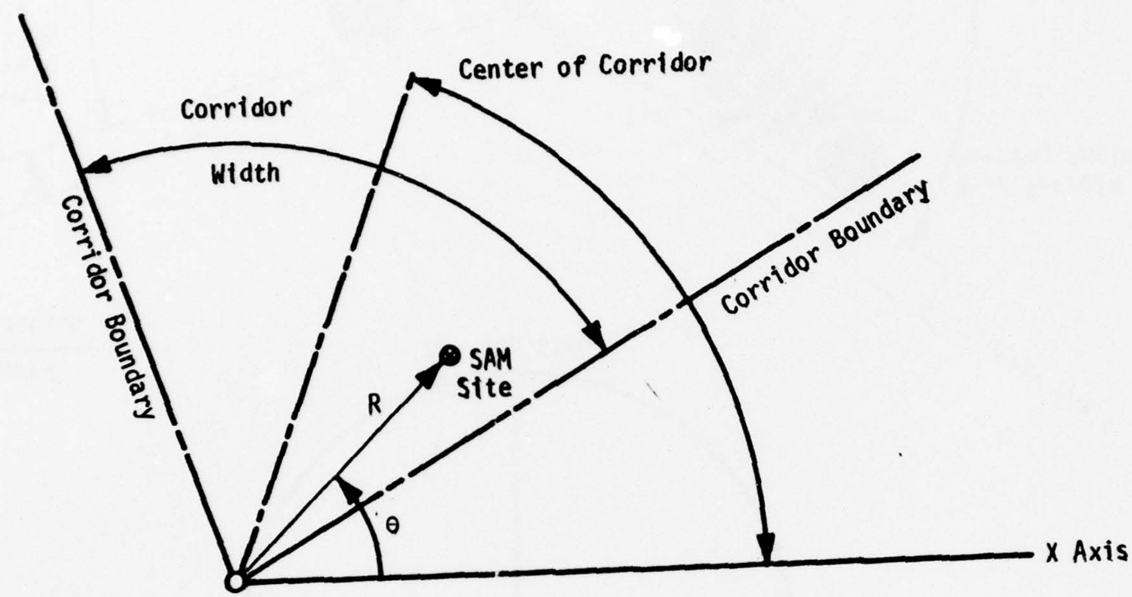


Figure 2. Polar Coordinates

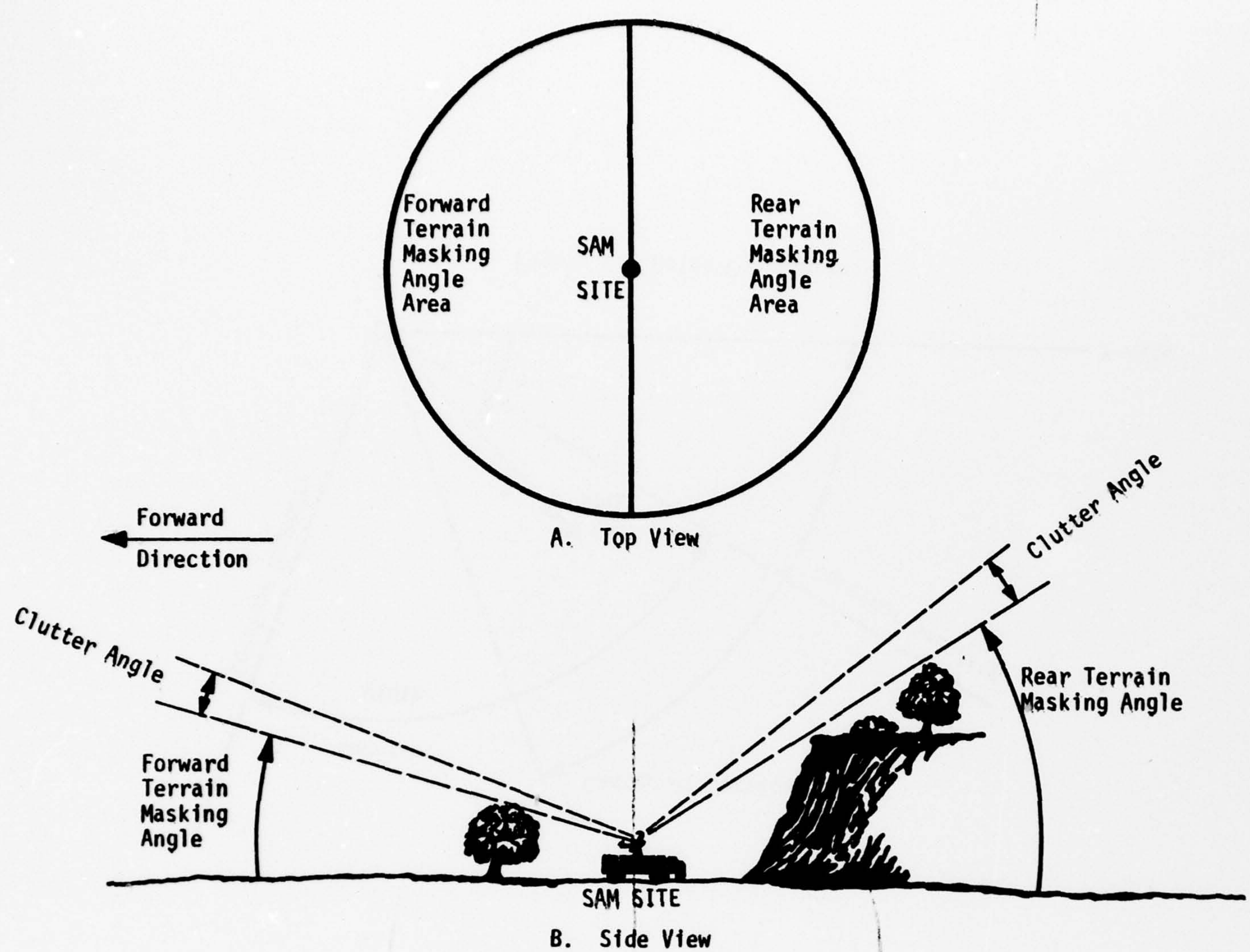


Figure 3. Terrain Masking Angles

the sensor height above ground level (ANTH - see TERRAIN), as well as terrain features (TFRAC - see TERRAIN). Up to 10 different series may be used to describe different sensor heights and terrain features. The problem is run for each pair of masking angles in a series. The results are weighed by the probability of occurrence to get an average for that series.

Rectangular coordinates are most useful in describing scenarios for enroute attrition over a large area. Polar coordinates, when used in the expected value mode, are convenient for certain types of terminal problems. For example, consider a target defended by several SAMs that is being attacked by a penetrator that may approach from any direction. This scenario is easily described in the polar coordinate system. The target is located at the origin and the SAMs at some radial distance from it. The penetrator profile is oriented to attack the target at the origin of the rectangular coordinate system. The model then simulates the different directions of attack by changing the positions of the SAMs around the target rather than by actually changing the penetrator profile. The probability of survival is thus averaged over all directions of attack. Two orientations may be selected for the forward direction of the SAM sites (AREA). Selecting the sites to always face radially outward from the origin simulates a penetrator approaching from any direction. Selecting the sites to always face in a negative X direction of the rectangular coordinate system simulates an unknown angular location of the sites around the target.

Once the set of SAM sites (SITE) has been specified, two ways are available for changing the relative number of SAMs in the set. The relative number of each SAM type may be altered by a multiplicative factor (DF). Additionally, it is often of interest to look at different multiples (overall defense levels) of the resulting set of SAM sites. The total number of all SAMs specified for the defense may be altered by a multiplicative factor (DLEV). A series of values may be specified and the probability of survival will be calculated for each value.

2. <u>Penetrator and Weapon</u>. Two separate flight paths may be handled by SURVIVE at the same time; those of a penetrator (XYZT) and the weapon it launches (XYZTW). The flight paths specify the time dependent position of the penetrator and weapon in three dimensions. X and Y are in the rectangular coordinate system described previously, and the third coordinate is altitude above the X-Y plane.

It is convenient to set up the problem such that the forward edge of the battle area (FEBA) is located along the Y axis with the area defended by the SAMs to the right of the Y axis. The SAM sites are assumed by the model to be set up facing in the negative X direction. Flight profiles would normally originate to the left of the FEBA and ingress to the defended area. (The profile could alternately be oriented to attack from behind the SAM sites.)

It is convenient when the model is used in the expected value mode to specify the flight paths relative to an imaginary target located at X and Y coordinates of zero. The flight paths are then shifted by an increment in X (XSTART) to position the target (and, hence, the flight path) at the desired location in the corridor. The resulting probabilities of survival for any given target location are highly dependent on the relative positions of the flight paths to the SAM locations. Sometimes it is desirable to have an average probability of survival for a target situated over a range of locations. The model will generate a series of flight paths starting with XSTART and incrementing it (DXSTART) to obtain a specific number of paths (NXSTART). The program then averages the probabilities of survival for all target locations.

SURVIVE can simulate SAM radar and IR sensor performance. The model has provisions for specifying radar and IR signatures for both the penetrator and weapon (SIGNATURE). The signatures are specified as a function of aspect angle around the penetrator or weapon (figure 4). The radar signatures are specified as the apparent radar cross-section area. The model determines the signal strength at the sensor as the apparent radar cross-section area divided by the distance between the target and the SAM site raised to the fourth power. The IR signatures are specified as radiated power. The signal strength at the sensor is radiated power divided by the distance between the target and the SAM site squared. Minimum signal strengths for sensor tracking (RTRK - see SAM) and for sensor lock-on (RLOCK - see SAM) must be specified.

Electronic countermeasures are not directly modeled in SURVIVE, but these effects are simulated by a degradation factor (ECME - see SITE) that reduces the effectiveness of the SAM sites by reducing the single shot kill probability. ECM degradation of the SAM sites may be applied selectively to different portions of the defended area by specifying intervals in X or R (XECM). All SAM

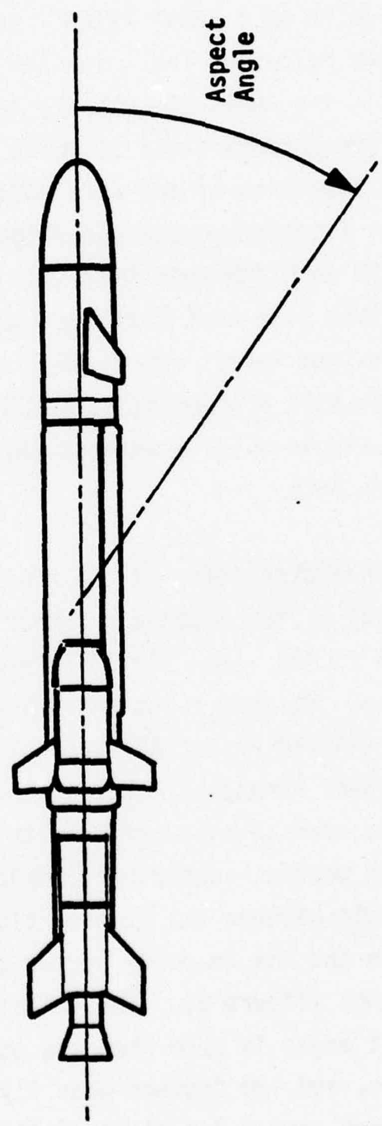


Figure 4. Signature Aspect Angle

sites with X or R coordinates within those intervals will be treated by the model as being subjected to ECM degradation.

3. SAM Characteristics. Ten different types of SAM sites may be modeled in SURVIVE (SAM). The characteristics of each SAM site type (number of missiles per salvo, number of salvos, etc.) are specified by input. Each type may be differentiated according to its firing capabilities, missile performance, sensor tracking, and missile launch characteristics. The firing capability of a SAM is described by the number of missiles it fires per salvo (NSS - see SAM), the time between firing each missile in a salvo (TISH - see SAM), the number of salvos that can be fired before reloading (NS - see SAM), the minimum time between successive salvos (TINTER - see SAM), the missile launcher reloading time (TRELOAD - see SAM), and the maximum azimuth angle at which missiles may be launched, measured from the forward direction of the site (AZMAX - see SAM). Missiles are assumed to fly straight line intercepts and flight times are specified by a time vs distance function for each SAM type (MISLXT). Missile intercept envelopes are described by a minimum intercept altitude (ALTMIN - see SAM), as well as dead zone (FUSE) and maximum lethal range (RNG) as a function of elevation angle (figure 5). Single shot kill probabilities (PKSS) are used to describe the average lethality of a single missile against the penetrator and the weapon, but need not be the same for both.

The geometric characteristics of the tracking system are described by the height above ground level of the antenna or other sensor (HRAD - see SAM) and its maximum elevation angle (ELMAX - see SAM). Several restrictions may be placed on the sensor's performance. Maximum range constraints may be used for specifying initiation of tracking (RADTRK - see SAM) and guidance system lock-on (RLOCK - see SAM), or minimum signal strengths for tracking (RTRK - see SAM) and lock-on (RLOCK - see SAM) can be used in conjunction with penetrator and weapon signatures to more accurately predict sensor performance. Restrictions may also be placed on the aspect angle between the line of sight vector from the SAM site to the penetrator or weapon and the velocity vector of the penetrator or weapon at the time of missile launch (figure 6). The velocity vector is tangent to the flight path. The aspect angle is zero when the penetrator or weapon is flying directly at the SAM site, and 180 degrees when flying directly away from it. The minimum and maximum aspect angles for firing (ASPMIN, ASPMAX - see SAM) allow the model to simulate systems that cannot lock on to targets with negative Doppler

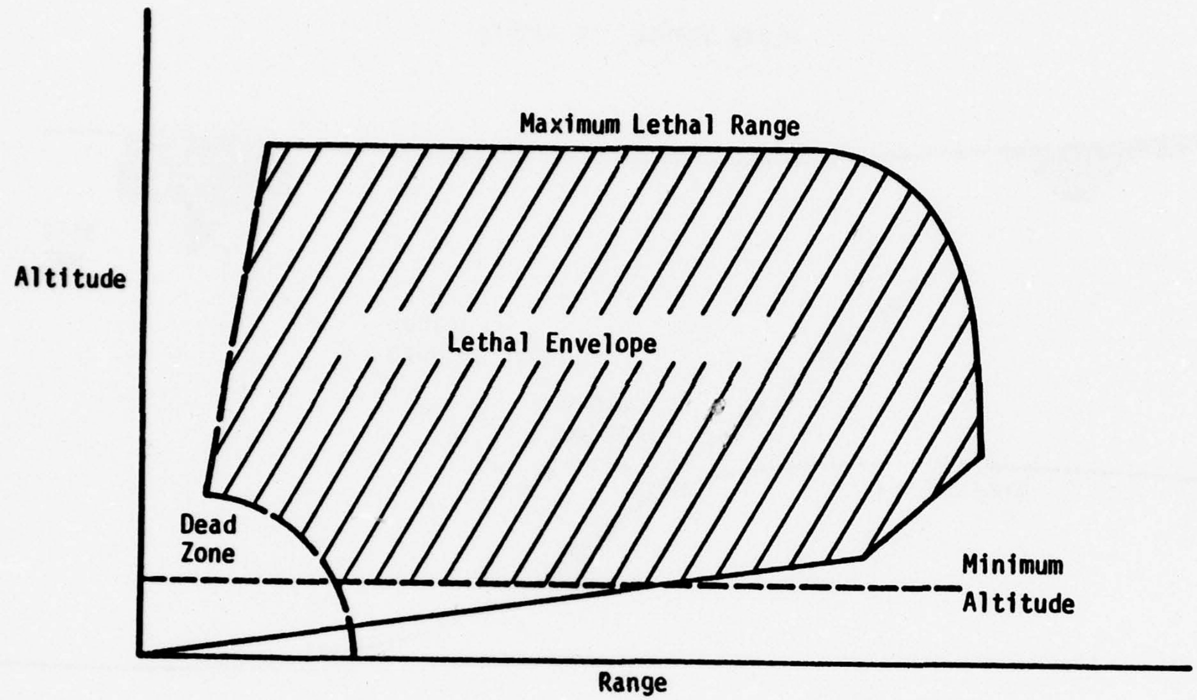


Figure 5. Missile Lethal Envelope

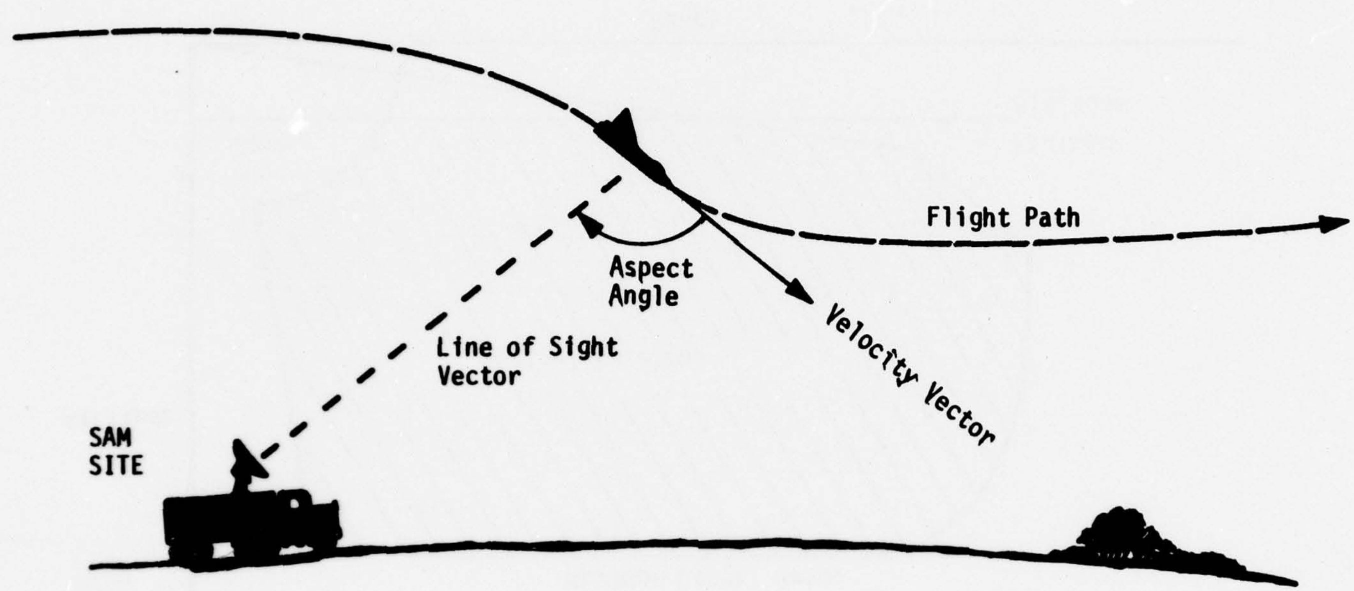


Figure 6. Aspect Angle

shifts, i.e., moving away from the SAM site. This feature can also be used to simulate an IR system in the absence of signature data because in most cases an IR sensor cannot lock on to a penetrator when it is headed toward the sensor, i.e., for small aspect angles, since it cannot see the heat source.

A look-shoot-look firing doctrine is employed for the SAMs. The SAM must be able to track the penetrator or weapon for a specified time (TINIT - see SAM) before it can fire a salvo. It must wait the same amount of time after the last missile of the salvo "intercepts" to evaluate the effects of the salvo and update its tracking information before it can fire again. This interval may not be less than the minimum time between salvos or the reload time, depending on the number of salvos the site has already fired. The site will fire as long as it is able to track and intercept the penetrator.

The model allows two types of missile guidance systems (IR - see SAM). The first is a guidance system that is self-homing, such as an IR seeker; the second must have the target in sensor coverage from the site to guide the missile to intercept.

It is possible that SAMs would not fire as readily at egressing penetrators, preferring to save their missiles for ingressing penetrators that might be more of a threat. The model can allow for this (EGRESS) by specifying a time (TEGRESS - see EGRESS) corresponding to the point on the penetrator flight path after which the SAMs firing will be less frequent, and a multiplicative factor (FEGRESS - see EGRESS) that will degrade the performance of the SAM sites after that time.

Two operational characteristics of the SAMs are site specific. When the locations and numbers of the sites are specified (SITE), the probability that the sites at each location will fire at a penetrator (PUP - see SITE) and the terrain identifying factor (TERFR - see SITE) for those sites must be specified.

C. CALCULATION OF AVERAGE NUMBER OF MISSILES FIRED AND SITE LETHAL WIDTH. The model simulates encounters between the SAMs and the penetrator and its weapon as a function of time to determine when missiles may be fired. Start and stop times corresponding to the part of the penetrator profile (TSTART, TSTOP) and weapon profile (TSTARTW, TSTOPW) to be considered in the survivability calculations

must be specified. The model will derive the simulation time interval limits (TSTARTP, TSTOPP) from the start and stop times of the penetrator and weapon. If a value of TSTARTP or TSTOPP is specified by input to the model, it will supersede the derived value. The model calculates both the average number of missiles fired and site lethal width when used in the expected value mode. When used in the specific mode, it calculates the number of missiles fired only from the specific site locations and does not vary the geometry between the sites and the penetrator path. The site lethal width is not calculated.

In calculating the lethal width of a SAM site and average number of missiles fired, the model considers only one SAM site at a time. Using the maximum range of the missile, the model determines the time intervals along the flight paths that the penetrator and weapon are in coverage of the site. It then finds the range in the Y direction from the site for which intercepts can occur. This range is then subdivided to give a representative number of different geometries by offsetting the flight path from the site in the Y direction. The model maps the flight paths and SAM sites into a spherical earth coordinate system to provide a more accurate representation of the problem geometry by accounting for the curvature of the earth. This mapping is accomplished by translating linear dimensions in the X-Y coordinates into arc lengths at the corresponding altitudes above the surface of the spherical earth. This provides a representation of horizon effects on the geometry between the SAM site and the flight paths. The model then examines the in-coverage time intervals using a small time step to determine the number of missiles the site can fire at the profiles for each offset. The lethal width of the site is calculated by multiplying the number of offsets with at least one missile firing by the offset subdivision interval. The number of missiles fired by the site are averaged over all offsets for which intercepts are possible. The model keeps track separately of the average number of missiles fired and site lethal width for the penetrator prior to weapons release, for the complete flight path, and for the weapon during its flight. The sites will continue to fire at the penetrator after the penetrator launches its weapon until a missile may be fired at the weapon. At this point, the site will guide any previously fired missiles to the penetrator (if the missiles are not self-homing) and starts tracking and firing at the weapon. The site will fire at the weapon throughout its flight. When the SAM site is no longer able to fire at the weapon, it will re-acquire the penetrator if it is in coverage and continue firing at it.

The calculation of the average number of missiles fired is quite time consuming, and the model makes several provisions for decreasing the calculation time at the expense of precision. One is a parameter for increasing the time step for examining the in-coverage intervals (FASTRUN). The value multiplies the normal timestep. A value of three will cut running time by about 50% and only changes the calculated value of survival probability by about 3%.

The other provision eliminates the need for calculating the average number of shots and lethal width for similar sites that cover identical portions of the flight profiles when the model is used in its expected value mode. On many types of profiles, a significant portion of the flight path will appear identical to any SAM site of a given type that can shoot only at that portion of the path. The model has an input parameter for each flight path that specifies the largest value of the X coordinate for this portion of its flight profile (XYZT, XYZTW). For example, an aircraft flight profile that ingresses at constant speed and altitude performs a maneuver and then egresses parallel to its ingress path at a constant speed and altitude would appear identical to a SAM site at any location up to the area in which the maneuver is performed. In figure 7 the flight path would appear identical to a SAM site located at a constant Y and at any X up to the point where it could interact with the penetrator during its maneuver. It is obvious that the average number of missiles fired and site lethal width for SAMs A, B, and C will be the same, so they need only be calculated once.

D. CALCULATION OF SURVIVAL PROBABILITY. When the model has calculated the average number of missiles fired by all SAM sites and their lethal widths, all the information necessary for calculating the survival probability is available. The probability of survival is calculated for each value of the series of defense levels (DLEV).

The probability of a single penetrator surviving the attack of type j SAMs at a site location i for defense level m is:

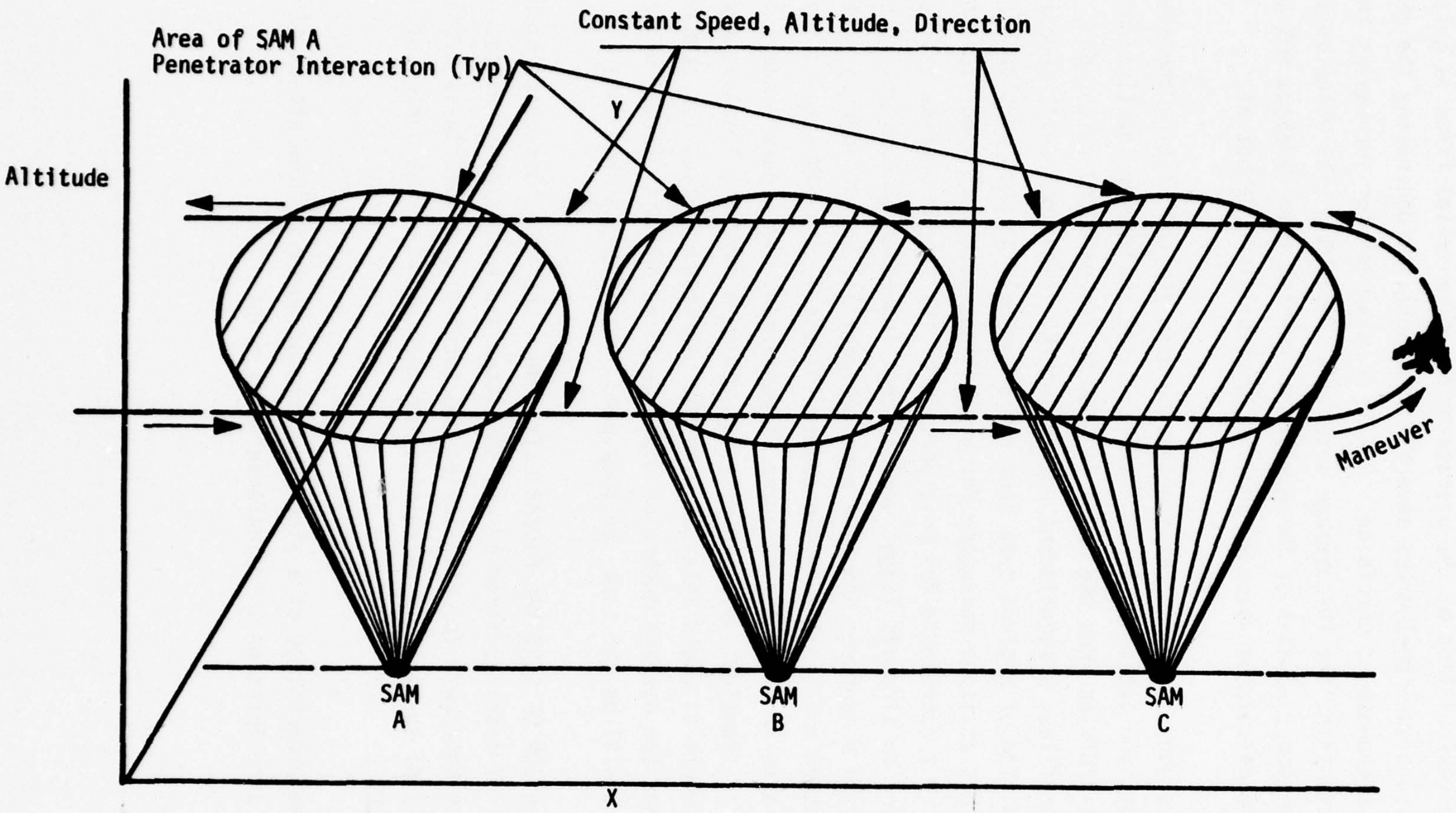


Figure 7. Portion of Flight Path That Appears Similar to SAMs at Different Locations

$$PS_{im} = \begin{cases} (1.-PK_j PECM_j)^{N_{im}}, & N_{im} > 1 \\ 1.-PK_j PECM_j N_{im}, & N_{im} \leq 1 \end{cases}$$

where

PK_j — single shot kill probability of the j type SAM

$PECM_j$ — effectiveness of SAM if subjected to ECM, unity otherwise.

The expected number of shots fired by the j type SAMs at location i for defense level m is:

$$N_{im} = (NSITE_i)\ (S_i)\ (DF_j)\ (PUP_i)\ (DLEV_m)\ (PE_i)$$

where

$NSITE_i$ — number of SAM sites at location i

S_i — number of missiles fired by a j type SAM at location i

DF_j — fraction of total number of j type SAMs operative

PUP_i — probability that a j type SAM site at i will fire at the penetrator, given the opportunity

$DLEV_m$ — defense level fraction

PE_i — expected number of encounters of penetrator with site at i

$$PE_i = \begin{cases} 1.0 & \text{specific placement of SAMs relative to target path} \\ L_i \div W & \text{expected value encounter with homogeneous boundary condition} \\ \left.\begin{array}{ll} 1.0 & L_i \geq 2W \\ \dfrac{L_i}{W} - \left(\dfrac{L_i}{2W}\right)^{2*} & L_i < 2W \end{array}\right\} & \text{expected value encounter with isolated corridor boundary condition} \end{cases}$$

where

L_i — lethal width of SAM at i

W — corridor width (CORWDTH)

*The factor $(L_i \div 2W)^2$ is the expected number of encounters with sites outside the boundaries of the corridor and must be subtracted when the isolated corridor boundary is used.

The probability of the penetrator surviving all the SAM locations for defense level m is:

$$P_m = \prod_{i=1}^{k} PS_{im}$$

where

k total number of SAM locations.

SECTION 3

INPUT AND OUTPUT

A. INPUT. Input to the SURVIVE program is based on cards containing an identifier and, generally, data associated with the identifier. The identifier may be the name of a variable in the model. The first ten spaces on the card are used for the identifier, left justified, and the second ten spaces are a floating point field for data. The remainder of the card may be used for comments. In many instances, this format is not adequate to allow the specification of all data associated with the identifier. In these cases, the identifier card is immediately followed by additional specially formatted data cards. The order of the identifier cards within a data deck is not important, although cards associated with an identifier card must be in proper order. A specific case is set up by defining all the necessary information for the model with the identifier cards and their associated cards. The case is terminated by the identifier "ENDCASE." Multiple cases may be run by the model. Once data are defined by an identifier case, they may be changed by specifying new data with another identifier card with the same identifier name. Data which does not change from case to case need only be input once. Additional cases may be defined by changing parameters of the previous data set, thus simplifying the generation of multiple cases. Program execution is immediately terminated by the identifier "ENDJOB." The program prints out all the input data to provide a permanent record of the parameters associated with each run. Data appearing on cards associated with the identifier card are labeled in the output with variable names or descriptions. These parameter names and descriptions were given in the previous section to show their use in the model. In this section they are used in addition to the identifier names to help describe the input to the model. The model checks for invalid identifier names and will issue a diagnostic message and terminate execution when one is found. Input cards for a sample problem are given in appendix B.

The following list defines the identifiers used for input to the model along with any other cards associated with the identifier. When an identifier card is used to specify a SAM type, the identifying number of the SAM from 1.0 to 10.0 is entered in the data field. In some cases, a variable number of cards may follow the identifier card, as when defining a flight profile. These groups of cards are terminated by an end of record (EOR) card. The model reads data

until the EOR card is reached, thus relieving the user of the task of counting the cards. The units used by the model are kilometers, seconds, degrees, and appropriately derived units. Parameter default values are given below when applicable.

Input Glossary

AREA — Signifies rectangular coordinates (= 1.), polar coordinates with site facing to decreasing values of X (= 0.), or polar coordinates with site facing radially outward from the origin of the coordinate system (= -1.). Default value = 1.

AVERAGE — Specific mode (= 0.), or expected value mode (= 1.). Default value = 1.

CLUTTER — Ground clutter angle above terrain for sensors.

CORWDTH — Corridor width.

DEBUG — Print debugging information concerning relative positions of sites and targets at each time step (= 1.), no information printed (= 0.). Default value = 0.

DEBUG1 — Print debugging information concerning relative positions of sites and targets for each missile launch (= 1.), no information printed (= 0.). Default value = 0.

DF — The i-th field of the next card contains the fraction of type i SAMs that are to constitute the actual defense level. Format (10E8.1).

DLEV — Each following card defines an overall defense level multiplier for editing the final output. Format (E10.3). Terminated by an EOR.

DXSTART — Increment in X for generating a series of delivery system profiles. Default value = 2.5.

EGRESS — The following card provides information on egress time of the penetrator and SAM degradation factor; TEGRESS the time after which the penetrator is assumed to be egressing and FEGRESS the degradation factor applied to SAM performance after that time. Format (2E10.3).

ENDCASE Signifies the end of input for a case.

ENDJOB Immediately terminates execution.

FASTRUN Multiplies the normal program time step to decrease running time. Default value = 1.

FUSE Specifies a SAM type. The following cards define the dead zone about the site by paired elevation-range points ordered by increasing elevation. Format (2E10.3). Terminated by an EOR.

MISLXT Specifies a SAM type. The following cards define the missile intercept performance by paired distance-time points along its flight path ordered by increasing distance. Format (2E10.3). Terminated by an EOR.

NOSIG The i-th field of the following card indicates that for SAM type i signature information and sensor sensitivities (= 0), or maximum sensor ranges (= 1) will be used for tracking and lock-on. Format (10I8). Default values = 1.

NXSTART Total number of flight profiles to be generated for this case. Default value = 1.

PKSS Selects the penetrator (= 0.) or the weapon (= 1.), and specifies that the single shot kill probabilities for the type i SAM against that target are given in the i-th field of the following card. Format (10E8.1).

RNG Specifies a SAM type. The following cards define the missile maximum lethal range envelope by paired elevation-range points ordered by increasing elevation. Format (2E10.3). Terminated by an EOR.

SAM Specifies a SAM type. The following three cards define various parameters for that type SAM site as follows:

Card 1: Format (3I10,5E10.3)

field (1)	NS	number of salvos that the site is able to fire before reloading.
(2)	NSS	number of missiles per salvo.
(3)	IR	missile guidance independent of site (= 1), target must remain within radar coverage of site during missile flight (= 0).
(4)	HRAD	height of sensor above ground level.
(5)	RTRK	minimum signal strength for sensor tracking.
(6)	RLOCK	minimum signal strength for sensor lock-on.
(7)	ELMAX	sensor maximum elevation.
(8)	ALTMIN	minimum missile intercept altitude

Card 2: Format (8E10.3)

field (1)	TINIT	sensor tracking time before each salvo is fired.
(2)	TISH	time between shots within a salvo.
(3)	TINTER	time between salvos.
(4)	TRELOAD	site reload time.
(5)	ASPMIN	minimum aspect angle of target for sensor to acquire penetrator or weapon.
(6)	ASPMAX	maximum aspect angle of target for sensor to acquire penetrator or weapon.
(7)	AZMAX	maximum azimuth angle for firing a missile.
(8)	ECME	effectiveness of site when subjected to ECM.

Card 3: Format (2E10.3)

field (1) RADTRK maximum sensor tracking range.

(2) RADLOCK maximum sensor lock-on range.

SIGNATURE Signifies signature data for (= 1.) the penetrator for use by those sensors with IR = 0, for (= 2.) the penetrator for use by those sensors with IR = 1, for (= 3.) the weapon for use by those sensors with IR = 0, and for (= 4.) the weapon for use by those sensors with IR = 1. The signature data is specified on the succeeding cards as a function of aspect angle in order of ascending angles. The angle is the first value on each card and the signature second. Format (2E10.3). Terminated by an EOR.

SITE The cards that follow define the entire basic set of SAMs available for the case, their number, locations, and some site specific parameters. Only one SAM type at one location may be specified per card, but more than one site may be specified at that location. Each card defines the following variables:

(1) NTYPE type number of SAM.

(2) NSITE number of sites.

(3) XSITE X or θ coordinate of sites.

(4) YSITE Y or R coordinate of sites. It is not necessary to specify Y or θ when running in the expected value mode (AVERAGE = 1.).

(5) PUP probability that the sites will fire given an opportunity.

(6) TERFR terrain identifying factor for selecting the proper terrain masking angle distribution (see TERRAIN).

Format (2I10, 4F10.3) terminated by EOR

SYMETRY — Signifies homogeneous boundary condition (= 1.0) or isolated area boundary condition (= 0.). Default value = 1.

TARGETY — The Y value of the ground target coordinate relative to the delivery system profile when AREA = 1., or the central angle of an angular corridor when AREA = 0. Only used when SYMETRY = 0. Default value = 0.

TERRAIN — Signifies that groups of discrete terrain masking angle information will follow. The first card of each of the groups following the TERRAIN card specifies TFRAC, the terrain masking angle identifying factor, and ANTH, the sensor height above ground level for which the masking angles were generated. Format (2E10.3). The remaining cards in each group specify the forward and rear masking angles, ELF and ELR, as well as the probability that they will occur, PROB. Format (3E10.3). A maximum of ten groups may be specified with up to 10 paired angles in each group. Each group is terminated by an EOR, and an additional EOR must appear after the last group. The model chooses the appropriate group of terrain masking angles to match the terrain identifying factor (TERFR) given on the site location cards (SITE), as well as the height of the sensor for the particular type of site (HRAD) as specified by the site characteristics (SAM). It is necessary to have groups of terrain masking angles for all resulting combinations of HRAD and TERFR.

TITLE — Any remark punched in the comment field of this card will be used to title the output.

TSTART — Starting time of the penetrator flight profile.

TSTARTP — Simulation starting time. If not specified, the model will choose the smaller of TSTART and TSTARTW.

TSTARTW — Starting time of the weapon flight profile.

TSTOP — Stopping time of the penetrator flight profile.

TSTOPP — Simulation stopping time. If not specified, the model will choose the larger of TSTOP and TSTOPW.

TSTOPW — Stopping time of the weapon flight profile.

XECM — Signifies that the following cards will specify the starting and ending values of intervals in X or R over which the SAM sites will be degraded by ECM, with one interval per card. Format (2E10.3). Terminated by an EOR.

XSTART — Constant added to the X coordinate of the penetrator and weapon flight profiles for locating it relative to the SAM sites. Will be the first value when generating a series of flight profiles for a given case.

XYZT — Specifies the largest X coordinate for the portion of the penetrator flight profile that will appear identical to similar SAM sites. A value of zero indicates that all SAM sites will be calculated individually. The following cards specify penetrator X, Y, Z, and corresponding time for each point of the flight profile in order of increasing time. Format (4E10.3). Terminated by an EOR.

XYZTW — The same as XYZT except for the weapon flight profile.

B. OUTPUT. After listing all data input to the model, survival probabilities and related information for each case are reported. Figure 8 shows part of the survival probability results for the sample problem in appendix B. Various features of the results will be identified in figure 8 as they are discussed in the text.

The model summarizes the results for each XSTART generated for a given case (A). For each of the overall defense levels input to the model, survival probabilities are reported without any ECM degradation of the SAM sites (C), as well as with any ECM degradation specified (D). Under each of these headings survival probabilities as a function of overall defense level are given for: (1) the penetrator from the simulation start time to the weapon start time (release) (E); (2) the penetrator from the simulation start time to the simulation stop time (F); and (3) the weapon from its start time to its stop time (G). In the last column (H) is given the combined weapon survival probability composed of the penetrator survival probability with ECM up to weapon release and the weapon

```
------------------------------------------------------------------------------------------------------------------------------
RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE
XSTART =        40.00
                    SURVIVAL PROBABILITY WITHOUT ECM                           SURVIVAL PROBABILITY WITH ECM
 DEFENCE LEVEL  A/C TO RELEASE   A/C COMPLETE      WEAPON/REL  A/C TO RELEASE   A/C COMPLETE      WEAPON/REL   A/C+W W/O ECM
         .250              .763           .664           .955            .866           .788            .955            .827
         .500              .571           .423           .911            .743           .609            .911            .677
         .750              .419           .270           .868            .631           .469            .868            .548
        1.000              .312           .174           .827            .539           .362            .827            .446
SITE    X OR TH       Y OR R LETH WDTH   AV SHOTS LETH WDTH   AV SHOTS LETH WDTH   AV SHOTS
                              A/C TO DELIVERY       A/C COMPLETE       WEAPON (REL TO TAR)
   1     5.000     0.000    19.000     2.842    23.000     3.152     0.000     0.000
   1    10.000     0.000    19.000     2.842    23.000     3.152     0.000     0.000
   1    25.000     0.000    19.000     2.842    23.000     3.152     0.000     0.000
   1    35.000     0.000    19.000     2.421    23.000     2.674    12.000     1.833
   1    45.000     0.000     1.000     1.000    16.500     2.139    19.000     2.664
   1    60.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   1    70.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   1    80.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000

******************************************************************************************************************************

AVERAGES FOR PREVIOUS     4 CASES   FIRST XSTART =      10.00
SAMPLE PROBLEM FOR SURVIVE MODEL
DEFINE SITE LOCATIONS
AIRCRAFT INGRESS, 180 DEGREE TURN, EGRESS
 WEAPON TRAJECTORY - CIRCULAR
AREA=  1.0 CORRODOR WIDTH     20.00 XSTART=      10.00
START TIMES  -1800.00       0.00  STOP TIMES    1829.00       40.00
                    SURVIVAL PROBABILITY WITHOUT ECM                           SURVIVAL PROBABILITY WITH ECM
 DEFENCE LEVEL  A/C TO RELEASE   A/C COMPLETE      WEAPON/REL  A/C TO RELEASE   A/C COMPLETE      WEAPON/REL   A/C+W W/O ECM
         .250              .838           .760           .961            .942           .895            .969            .905
         .500              .696           .565           .922            .888           .801            .938            .819
         .750              .574           .429           .884            .837           .720            .907            .740
        1.000              .479           .329           .847            .793           .653            .879            .672
******************************************************************************************************************************
```

Figure 8. Sample of SURVIVE Results

survival probability without ECM. The site lethal width and average number of missiles fired for a single SAM site that are used to calculate the survival probabilities are given by SAM type and location (I).

After these results have been given for all values of XSTART generated for the case, the average survival probabilities for the case are given (B).

APPENDIX A

FORTRAN LISTING OF SURVIVE

```
      PROGRAM SURVIVE(INPUT,OUTPUT,TAPE1=INPUT)
C
C
C     THIS ROUTINE SEQUENCES THE PROGRAM FLOW AND PRINTS CASE AVERAGES
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),PLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      COMMON /AVG/ SUMECM(10,3),SUMNECM(10,3)
    1 CALL INPUTS
      DO 2 I=1,30
    2 SUMECM(I)=SUMNECM(I)=0.
      IF (NXSTART.LE.0) NXSTART=1
      SVXS=XSTART
      DO 3 I=1,NXSTART
      CALL AVSHOTS
      CALL PROBS
      XSTART=XSTART+DXSTART
    3 CONTINUE
      XSTART=SVXS
      PRINT 4, NXSTART,XSTART
      PRINT 5, TITLE,NITL1,NITL2,NITL3
      PRINT 6, AREA,CORWDTH,XSTART,TSTART,TSTOP
      PRINT 8
      PRINT 7, (DLEV(L),(SUMNECM(L,I),I=1,3),(SUMECM(L,I),I=1,3),SUMECM(
     1L,1)*SUMNECM(L,3),L=1,NDLEV)
      PRINT 9
      GO TO 1
C
    4 FORMAT (//130(1H*)//22H AVERAGES FOR PREVIOUS,I5,22H CASES  FIRST
     1XSTART =,F10.2)
    5 FORMAT (1X,6A10)
    6 FORMAT (6H AREA=,F5.1,15H CORRIDOR WIDTH,F10.2,8H XSTART=,F10.2/12
     1H START TIMES,2F10.2,12H  STOP TIMES,2F10.2)
    7 FORMAT (8F15.3)
    8 FORMAT (21X,32HSURVIVAL PROBABILITY WITHOUT ECM,23X,29HSURVIVAL PR
     1OBABILITY WITH ECM/2X,13HDEFENCE LEVEL,2(1X,14HA/C TO RELEASE,3X,1
     22HA/C COMPLETE,5X,10HWEAPON/REL)2X,13HA/C+W W/O ECM)
    9 FORMAT (130(1H*))
      END
```

```
      SUBROUTINE AVSHOTS
C
C
C     THIS ROUTINE DETERMINES OFFSET RANGES, CALCULATES NUMBER OF SHOTS
C     FIRED, AND THE AVERAGE NUMBER OF SHOTS AND LETHAL WIDTH
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ISAM/ NTYPE(100),NSITE(100),XSITE(100),PUP(100),ELMIN(100,
     12),NTOTS,SITWDTH(100,3),AVSHOT(100,3),TERFR(100),SITRAD(100),ARSIT
     2(100)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      COMMON /D/ AZT,ELT,RNGT,ASPT,AZTF,ELTF,RNGTF,ASPTF,AZTA,ELTA,RNGTA
     1,ASPTA
      DIMENSION NDONE(100), NSHT(3), AVSH(3), NOFSET(3)
      DO 1 I=1,100
    1 NDONE(I)=0
      DPR=57.29578
      NSDT=FASTRUN-1.
      IF (NSDT.LT.0) NSDT=0
      NDEG=10
      NDOFS=10
      NDTS=50
      TINT=15.
      SVT1=TSTARTP
      SVT2=TSTOPP
      DO 45 I=1,NTOTS
      TSTARTP=SVT1
      TSTOPP=SVT2
      IF (NDONE(I).GT.0) GO TO 45
      XSITS=XSITE(I)
      N=NTYPE(I)
      SITWDTH(I,1)=SITWDTH(I,2)=SITWDTH(I,3)=0.
      AVSHOT(I,1)=AVSHOT(I,2)=AVSHOT(I,3)=0.
      IF (DF(N).LE.0.) GO TO 45
      IF (NTER.LE.0) GO TO 45
      IS(1)=IS(2)=2
      DT=5.
      NSYSOP=0
      IF (NXYZT(1).GT.0.A.PKSS(N,1).GT.0..A.(IR(N).LE.0.A.NSIG(1).GT.0.O
     1.IR(N).GT.0.A.NSIG(3).GT.0.O.NINTRPR(N).GT.0)) NSYSOP=1
      IF (NXYZT(2).GT.0.A.PKSS(N,2).GT.0..A.(IR(N).LE.0.A.NSIG(2).GT.0.O
```

```
 1.IR(N).GT.0.A.NSIG(3).GT.0.O.NINTRPR(N).GT.0)) NSYSOP=NSYSOP+2
  IF (NSYSOP.LE.0) GO TO 45
  IF (TSTARTP.GT.-100000.) GO TO 5
  GO TO (2,3,4), NSYSOP
2 TSTARTP=TSTART
  GO TO 5
3 TSTARTP=TSTART(2)
  GO TO 5
4 TSTARTP=AMIN1(TSTART(1),TSTART(2))
5 IF (TSTOPP.GT.-100000.) GO TO 9
  GO TO (6,7,8), NSYSOP
6 TSTOPP=TSTOP
  GO TO 9
7 TSTOPP=TSTOP(2)
  GO TO 9
8 TSTOPP=AMAX1(TSTOP(1),TSTOP(2))
9 CONTINUE
  XSAME=XSAMEP
  IF (NSYSOP.EQ.2) XSAME=XSAMEW
  CALL TIMEIN (I,DT,N)
  IF (NTIN.LT.1) GO TO 45
  IF (AREA.LE.0.) GO TO 15
  GO TO (10,11,12), NSYSOP
10 FLTW1=FLTWTH(1,1)
  FLTW2=FLTWTH(2,1)
  GO TO 13
11 FLTW1=FLTWTH(1,2)
  FLTW2=FLTWTH(2,2)
  GO TO 13
12 FLTW1=AMAX1(FLTWTH(1,1),FLTWTH(1,2))
  OFMAX=-FLTW2+RLETH(N)
13 OFMIN=-FLTW1-RLETH(N)
  OFMAX=FLTW2+RLETH(N)
  OFMIN2=-FLTWTH(1,2)-RLETH(N)
  OFMAX2=-FLTWTH(2,2)+RLETH(N)
  IF (SYMETRY.GT.0.) GO TO 14
  OFMAX=AMIN1(TARGETY+CORWDTH,OFMAX)
  OFMIN=AMAX1(TARGETY-CORWDTH,OFMIN)
14 CONTINUE
  DOFS=RLETH(N)/NDOFS
  NOF=(OFMIN-DOFS)/DOFS
  OFMIN=DOFS*NOF
  NOF=(OFMAX+DOFS)/DOFS
  OFMAX=DOFS*NOF
15 CONTINUE
  DO 16 NNT=1,NTER
  IF (TERFR(I).EQ.TFRAC(NNT).A.HRAD(N).EQ.ANTH(NNT)) GO TO 17
16 CONTINUE
  GO TO 45
17 NALF=NNT
  NNT=NTERA(NALF)
  DO 39 NNT1=1,NNT
  ELMIN(I,1)=TERRANE(NNT1,1,NALF)+CLUTTER
  ELMIN(I,2)=TERRANE(NNT1,2,NALF)+CLUTTER
```

```
      AVSH(1)=AVSH(2)=AVSH(3)=0
      NOFSET(1)=NOFSET(2)=NOFSET(3)=0
      OFSET=OFMIN
      IF (AVERAGE.LE.0.) OFSET=-SITRAD(I)
      LAB=10HSAM
      IF (DEBUG1.GT.0.) PRINT 46, LAB,FLOAT(N),FLOAT(I),OFMIN,OFMAX,ELMI
     1N(I,1),ELMIN(I,2),FLOAT(NSYSOP),TSTARTP,TSTOPP
   18 NSHO=0
      ISHOT=0
      NSR=0
      NSHT(1)=NSHT(2)=NSHT(3)=0
      TLW=TSTARTP
      TS=TSTARTP
      TOL=TS
      NPH=1
      IF (NSYSOP.EQ.2) NPH=3
      NSF=0
      DO 36 NT=1,NTIN
      IF (AREA.GT.0.) GO TO 21
      OFMIN=THIN(NT,1)
      OFMAX=THIN(NT,2)
      DOFS=IFIX(ARSIT(I)/NDEG)
      DOFS=AMIN1(10.,DOFS)
      DOFS=AMAX1(1.,DOFS)
      IF (OFMAX-OFMIN.GE.359.9) OFMAX=OFMAX-.5*DOFS
      OFSETD=OFMIN
      IF (AVERAGE.LE.0.) OFSETD=XSITS
   19 XSITE(I)=SITRAD(I)*COS(OFSETD/DPR)
      OFSET=-SITRAD(I)*SIN(OFSETD/DPR)
      NSHT(1)=NSHT(2)=NSHT(3)=0
      TLW=TSTARTP
      TS=TSTARTP
      TOL=TS
      NPH=1
      NSF=0
      NSHO=0
      IF (SYMETRY.GT.0.) GO TO 20
      TEMPD=AMOD(OFSETD+720.,360.)
      TEMP1=AMOD(TARGETY-.5*CORWDTH+720.,360.)
      TEMP2=AMOD(TARGETY+.5*CORWDTH+720.,360.)
      IF (TEMP2.GT.TEMP1.A.(TEMPD.LT.TEMP1.O.TEMPD.GT.TEMP2)) GO TO 34
      IF (TEMP2.LT.TEMP1.A.(TEMPD.LT.TEMP1.A.TEMPD.GT.TEMP2)) GO TO 34
   20 CONTINUE
   21 CONTINUE
      TO=TIN(NT,1)
      DTIS=IFIX(.25*TISH(N))
      DTIS=AMAX1(DTIS,1.)
      DT=IFIX(AMIN1((TIN(NT,2)-TO)/NDTS,.25*TINTER(N),.25*TINIT(N)))
      DT=AMAX1(DT,1.)
      TO=TO-DT
      TINT=TINIT(N)
   22 IF (NSF.EQ.0) TO=TO+DT
      IF (NSF.GT.0) TO=TO+DTIS
      IF (TO.GE.TS) GO TO 23
```

```
      NNNT=(TS-TO)/DT+1
      TO=TO+NNNT*DT
   23 CONTINUE
      IF (FASTRUN.LE.1.) GO TO 25
      IF (ISHOT.NE.0) GO TO 24
      NSR=NSR+1
      IF (NSR.LE.NSDT) GO TO 25
      IF (NSF.EQ.0) TO=TO+NSDT*DT
      IF (NSF.GT.0) TO=TO+NSDT*DTID
      GO TO 25
   24 IF (NSR.GT.NSDT.A.NSF.EQ.0) TO=TO-NSDT*DT-DT
      IF (NSR.GT.NSDT.A.NSF.GT.0) TO=TO-NSDT*DTIS-DTIS
      NSR=0
   25 CONTINUE
      IF (TO.GT.TIN(NT,2)+.5*DT) GO TO 29
      IF (TS.GE.TIN(NT,2)) GO TO 34
      IF (OFSET.LT.OFMIN2.O.OFSET.GT.OFMAX2) GO TO 26
      GO TO (26,27,28), NSYSOP
   26 L=1
      GO TO 31
   27 L=2
      GO TO 31
   28 L=1
      IF (NPH.EQ.4) GO TO 31
      IF (TO.LE.TSTART(2)+TINIT(N)) GO TO 31
      IF (TO.GT.TIN(NT,2).O.TO.GT.TSTOP(2).O.TS.GE.TIN(NT,2).O.TS.GE.TST
     1OP(2)) GO TO 29
      L=2
      IF (NPH.EQ.3) GO TO 31
      NPH=2
      L=1
      CALL INCOV (I,TO,OFSET,ISHOT,TFIRE,TINT,TAQ,TSTART(2),2)
      IF (NSR.GT.NSDT.A.(NSDT+1)*DT+TO.GT.TSTOP(2).A.ISHOT.LE.0.A.FASTRU
     1N.GT.1.) NSR=0
      IF (ISHOT.LE.0) GO TO 31
      LAB=10HGOOD SHWEA
      IF (DEBUG1.GT.0.) PRINT 47, LAB,FLOAT(I),FLOAT(NPH),FLOAT(NSF),TFI
     1RE,TO,TS,TOL,TLW
      IF (TAQ.LT.TSTART(2)) GO TO 31
      IF (NSR.GT.NSDT) GO TO 22
      TLW=TO
      NPH=3
      L=2
      IF (IR(N).NE.0.A.NSF.EQ.0.A.TFIRE.GE.TS) GO TO 32
      IF (NSF.NE.0) TS=TS-TISH(N)+TINIT(N)
      IF (IR(N).NE.0) GO TO 31
      TS=AMAX1(TS,TOL+TINIT(N))
      GO TO 31
   29 IF (NSYSOP.NE.3) GO TO 34
      IF (NPH.EQ.1) GO TO 34
      IF (NPH.EQ.2) GO TO 30
      IF (NPH.EQ.4) GO TO 34
      TO=TLW
      IF (NSHT(3).GT.0.A.NSF.GT.0) TS=TS-TISH(N)+TINIT(N)
```

```
      IF (IR(N).EQ.0) TS=AMAX1(TS,TOL+TINIT(N))
      LAB=10HCHANGE 1
      IF (DEBUG1.GT.0.) PRINT 47, LAB,TS,TO,AVSH(3)
   30 L=1
      NPH=4
   31 CALL INCOV (I,TO,OFSET,ISHOT,TFIRE,TINT,TAQ,TS,L)
   32 IF (ISHOT.LE.0) GO TO 22
      IF (NSR.GT.NSDT) GO TO 22
      TOL=TO
      IF (AREA.GT.0.) OFSETD=OFSET
      LAB=10HGOOD SHOT
      IF (DEBUG1.GT.0.) PRINT 47, LAB,OFSETD,TS,TAQ,AZTA,ELTA,RNGTA,ASPT
     1A,TFIRE,AZTF,ELTF,RNGTF,ASPTF,TO,AZT,ELT,RNGT,ASPT,NPH
      IF (NPH.NE.3.A.NSYSOP.NE.2) AVSH(2)=AVSH(2)+1.
      IF (NPH.NE.3.A.NSYSOP.NE.2) NSHT(2)=1
      IF (NPH.NE.3.A.NSYSOP.NE.2.A.TAQ.GT.TEGRESS) AVSH(2)=AVSH(2)+FEGRE
     1SS-1.
      IF (NPH.NE.3.A.NSYSOP.NE.2.A.TO.LE.TSTART(2)) AVSH(1)=AVSH(1)+1.
      IF (NPH.NE.3.A.NSYSOP.NE.2.A.TO.LE.TSTART(2)) NSHT(1)=1
      IF (NPH.NE.3.A.NSYSOP.NE.2.A.TO.LE.TSTART(2).A.TAQ.GT.TEGRESS) AVS
     1H(1)=AVSH(1)+FEGRESS-1.
      IF (NSYSOP.NE.1.A.NPH.EQ.3.A.TAQ.GT.TSTART(2)) AVSH(3)=AVSH(3)+1.
      IF (NSYSOP.NE.1.A.NPH.EQ.3.A.TAQ.GT.TSTART(2)) NSHT(3)=1
      NSF=NSF+1
      TS=TFIRE+TISH(N)
      IF (NSF.LT.NSS(N)) GO TO 22
      NSF=0
      TS=TFIRE+TINTER(N)
      NSHO=NSHO+1
      IF (TS.GE.TO+TINIT(N)) GO TO 33
      TS=TO+TINIT(N)
   33 CONTINUE
      IF (NSHO.LT.NS(N)) GO TO 22
      NSHO=0
      TS=TFIRE+TRELOAD(N)
      IF (TS.GE.TO+TINIT(N)) GO TO 22
      TS=TO+TINIT(N)
      GO TO 22
   34 CONTINUE
      IF (AREA.GT.0.) GO TO 36
      DO 35 K=1,3
   35 IF (NSHT(K).GT.0) NOFSET(K)=NOFSET(K)+1
      LAB=10HPOLARLIMS
      IF (DEBUG2.GT.0.) PRINT 46, LAB,FLOAT(NT),DOFS,OFSETD,XSITE(I),OFS
     1ET,AVSH,FLOAT(NOFSET),TSTART
      IF (AVERAGE.LE.0.) GO TO 36
      OFSETD=OFSETD+DOFS
      IF (OFSETD.LE.OFMAX) GO TO 19
   36 CONTINUE
      IF (AREA.LE.0.) GO TO 38
      DO 37 K=1,3
   37 IF (NSHT(K).GT.0) NOFSET(K)=NOFSET(K)+1
      IF (AVERAGE.LE.0.) GO TO 38
      OFSET=OFSET+DOFS
```

```
      IF (OFSET.LE.OFMAX) GO TO 18
   38 CONTINUE
      DO 39 K=1,3
      IF (NOFSET(K).GT.0) AVSH(K)=AVSH(K)/NOFSET(K)
      SITWDTH(I,K)=NOFSET(K)*DOFS*TERRANE(NNT1,3,NALF)+SITWDTH(I,K)
      AVSHOT(I,K)=AVSHOT(I,K)+TERRANE(NNT1,3,NALF)*AVSH(K)
      LAB=10HAVSHOTS
      IF (DEBUG1.GT.0.) PRINT 46, LAB,FLOAT(I),FLOAT(NOFSET(K)),AVSHOT(I
     1,K),SITWDTH(I,K),AVSH(K),NOFSET(K)*DOFS
   39 CONTINUE
      XSITE(I)=XSITS
      DO 40 K=1,3
   40 IF (AVERAGE.LE.0.) SITWDTH(I,K)=CORWDTH
      IF (I.EQ.NTOTS.O.AREA.LE.0..O.XSAME.EQ.0.) GO TO 45
      K=1
      IF (NSYSOP.EQ.2) K=2
      XDONE=XSAME+XSTART-RLETH(N)
      IF (XSITE(I).GT.XDONE) GO TO 45
      XTSTOP=TRP(TSTOP,T(1,K),XYZ(1,1,K),NXYZT(K))+XSTART
      XTSTART=TRP(TSTART,T(1,K),XYZ(1,1,K),NXYZT(K))+XSTART
      XTEGR=TRP(TEGRESS,T(1,K),XYZ(1,1,K),NXYZT(K))+XSTART
      IF (ABS(XTEGR-XSITE(I)).LT.RLETH(N).A.FEGRESS.NE.1.) GO TO 45
      D3I=XTEGR-XSITE(I)
      IF (NSYSOP.EQ.2) GO TO 41
      XTREL=TRP(TSTART(2),T(1,K),XYZ(1,1,K),NXYZT(K))+XSTART
      IF (ABS(XTREL-XSITE(I)).LT.RLETH(N)) GO TO 45
      D4I=XTREL-XSITE(I)
   41 CONTINUE
      IF (ABS(XTSTART-XSITE(I)).LT.RLETH(N).O.ABS(XTSTOP-XSITE(I)).LT.RL
     1ETH(N)) GO TO 45
      D1I=XTSTART-XSITE(I)
      D2I=XTSTOP-XSITE(I)
      J1=I+1
      DO 44 J=J1,NTOTS
      IF (NTYPE(I).NE.NTYPE(J).O.TERFR(I).NE.TERFR(J).O.XSITE(J).GT.XDON
     1E.O.NDONE(J).GT.0) GO TO 44
      IF (ABS(XTSTART-XSITE(J)).LT.RLETH(N).O.ABS(XTSTOP-XSITE(J)).LT.RL
     1ETH(N)) GO TO 44
      D1J=XTSTART-XSITE(J)
      D2J=XTSTOP-XSITE(J)
      D3J=XTEGR-XSITE(J)
      IF (FEGRESS.NE.1..A.SGN(D3I).NE.SGN(D3J)) GO TO 44
      IF (NSYSOP.EQ.2) GO TO 42
      D4J=XTREL-XSITE(J)
      IF (SGN(D4I).NE.SGN(D4J)) GO TO 44
   42 CONTINUE
      IF (SGN(D1I).NE.SGN(D1J).O.SGN(D2I).NE.SGN(D2J)) GO TO 44
      DO 43 K=1,3
      SITWDTH(J,K)=SITWDTH(I,K)
      AVSHOT(J,K)=AVSHOT(I,K)
   43 CONTINUE
      NDONE(J)=1
   44 CONTINUE
   45 CONTINUE
```

```
      TSTARTP=SVT1
      TSTOPP=SVT2
      RETURN
C
   46 FORMAT (1X,A10,12F10.4)
   47 FORMAT (1X,A10,17F7.1,I3)
      END
```

```
      SUBROUTINE PROBS
C
C
C     THIS ROUTINE CALCULATES AND PRINTS PROBABILITY OF SURVIVAL FOR
C     EACH XSTART
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ISAM/ NTYPE(100),NSITE(100),XSITE(100),PUP(100),ELMIN(100,
     12),NTOTS,SITWDTH(100,3),AVSHOT(100,3),TERFR(100),SITRAD(100),ARSIT
     2(100)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      COMMON /AVG/ SUMECM(10,3),SUMNECM(10,3)
      DIMENSION PECM(10,3), PNECM(10,3)
      PRINT 14
      IF (AREA.LE.0.) PRINT 17
      IF (AREA.LT.0.) PRINT 18
      IF (AVERAGE.GT.0.) PRINT 15
      PRINT 16, XSTART
      DO 1 I=1,30
    1 PECM(I)=PNECM(I)=1.
      DO 12 I=1,3
      I1=1
      IF (I.EQ.3) I1=2
      DO 12 J=1,NTOTS
      N=NTYPE(J)
      IF (DF(N).LE.0.) GO TO 12
      ENC=1.
      IF (AVERAGE.LE.0.) GO TO 6
      IF (SYMETRY.LE.0.) GO TO 2
      ENC=SITWDTH(J,I)/CORWDTH
      GO TO 6
    2 IF (AREA.LE.0.) GO TO 4
    3 ENC=SITWDTH(J,I)/CORWDTH-(AMIN1(1.,.5*SITWDTH(J,I)/CORWDTH))**2
      GO TO 5
    4 IF (SITWDTH(J,I).GE.360..O.SITWDTH(J,I)/CORWDTH.GE.2.) GO TO 6
      IF (CORWDTH+.5*SITWDTH(J,I).LE.360.) GO TO 3
      A1=CORWDTH+.5*SITWDTH(J,I)-360.
      A2=CORWDTH-.5*SITWDTH(J,I)
      P1=(CORWDTH+SITWDTH(J,I)-360.)/CORWDTH
      ENC=(SITWDTH(J,I)+(A1+A2)*(P1-SITWDTH(J,I)/CORWDTH))/CORWDTH
    5 ENC=AMIN1(1.,ENC)
```

```
    6 EI=NSITE(J)*ENC
      SI=EI*AVSHOT(J,I)*PUP(J)*DF(N)
      ECMEFF=1.
      IF (NXECM.LE.0) GO TO 8
      DO 7 L=1,NXECM
      IF (XSITE(J).GE.XECM(L,1).A.XSITE(J).LE.XECM(L,2)) ECMEFF=ECME(N)
    7 CONTINUE
    8 CONTINUE
      DO 11 L=1,NDLEV
      IF (SI*DLEV(L).LE.1.) GO TO 9
      PSIE=(1.-PKSS(N,I1)*ECMEFF)**(SI*DLEV(L))
      PSI=(1.-PKSS(N,I1))**(SI*DLEV(L))
      GO TO 10
    9 PSIE=1.-PKSS(N,I1)*ECMEFF*SI*DLEV(L)
      PSI=1.-PKSS(N,I1)*SI*DLEV(L)
   10 CONTINUE
      PECM(L,I)=PECM(L,I)*PSIE
      PNECM(L,I)=PNECM(L,I)*PSI
   11 CONTINUE
   12 CONTINUE
      PRINT 19
      PRINT 20, (DLEV(L),(PNECM(L,I),I=1,3),(PECM(L,I),I=1,3),PECM(L,1)*
     1PNECM(L,3),L=1,NDLEV)
      PRINT 21
      PRINT 22, (NTYPE(I),XSITE(I),SITRAD(I),(SITWDTH(I,J),AVSHOT(I,J),J
     1=1,3),I=1,NTOTS)
      DO 13 I=1,3
      DO 13 L=1,NDLEV
      SUMECM(L,I)=SUMECM(L,I)+PECM(L,I)/NXSTART
      SUMNECM(L,I)=SUMNECM(L,I)+PNECM(L,I)/NXSTART
   13 CONTINUE
      RETURN
C
C
   14 FORMAT (//,1X,130(1H-))
   15 FORMAT (45H RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE)
   16 FORMAT (9H XSTART =,F10.2)
   17 FORMAT (16H RADIAL SURVIVAL)
   18 FORMAT (13H PATH AVERAGE)
   19 FORMAT (21X,32HSURVIVAL PROBABILITY WITHOUT ECM,23X,29HSURVIVAL PR
     1OBABILITY WITH ECM/2X,13HDEFENCE LEVEL,2(1X,14HA/C TO RELEASE,3X,1
     22HA/C COMPLETE,5X,10HWEAPON/REL)2X,13HA/C+W W/O ECM)
   20 FORMAT (8F15.3)
   21 FORMAT (1X,4HSITE,3X,7HX OR TH,4X,6HY OR R,3(1X,9HLETH WDTH,2X,8HA
     1V SHOTS),/27X,15HA/C TO DELIVERY,7X,12HA/C COMPLETE,4X,19HWEAPON (
     2REL TO TAR))
   22 FORMAT (I5,8F10.3)
      END
```

```
      SUBROUTINE INPUTS
C
C
C     THIS ROUTINE READS ALL INPUT DATA AND TERMINATES THE PROGRAM
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ISAM/ NTYPE(100),NSITE(100),XSITE(100),PUP(100),ELMIN(100,
     12),NTOTS,SITWDTH(100,3),AVSHOT(100,3),TERFR(100),SITRAD(100),ARSIT
     2(100)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      DIMENSION NTEP(6)
      INTEGER TITLE
      DATA DEBUG/0./,DEBUG1/0./,DF/10*0./,NTOTS/0/,NXYZT/2*0/,XSTART/0./
     1,TSTART/2*0./,TSTOP/2*0./,NXECM/0/,CLUTTER/.25/,NTER/0/,AREA/1./,D
     2EBUG2/0./
      DATA DXSTART/2.5/,NXSTART/1/
      DATA DLEV/1.,9*0./,NDLEV/1/,NSIG/4*0/,PKSS/20*0./,TSTARTP,TSTOPP/2
     1*-100000./
      DATA AVERAGE/1./
      DATA SYMETRY/1./
      DATA FASTRUN/0./
      DATA NITLI/12*10H          /,NITL3/6*10H          /
      DATA TARGETY/0./
      DATA TEGRESS/100000./,FEGRESS/1./
      DATA RADTRK/10*0./,RADLOCK/10*0./,NINTRPR/10*1/
      IER=0
      DATA TITLE/6*10H          /
      DATA NITL2/6*10H          /
      PRINT 60
    1 READ 61, N,X,NTEP
      PRINT 62, N,X,NTEP
      IF (N.NE.7HENDCASE) GO TO 2
      IF (IER.EQ.0) RETURN
      STOP
    2 IF (N.EQ.6HENDJOB) STOP
      IF (N.NE.5HTITLE) GO TO 4
      DO 3 J=1,6
    3 TITLE(J)=NTEP(J)
      GO TO 1
    4 IF (N.NE.4HXYZT) GO TO 8
      DO 5 J=1,6
```

```
    5 NITL2(J)=NTEP(J)
      XSAMEP=X
      PRINT 63
      K=NXYZT(1)=0
    6 READ 64, (XYZ(K+1,J,1),J=1,3),T(K+1,1)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 64, (XYZ(K+1,J,1),J=1,3),T(K+1,1)
      NXYZT(1)=K=K+1
      IF (K.EQ.1) GO TO 6
      DO 7 J=1,3
    7 DXYZDT(K,J,1)=(XYZ(K,J,1)-XYZ(K-1,J,1))/(T(K,1)-T(K-1,1))
      GO TO 6
    8 IF (N.NE.6HTSTART) GO TO 9
      TSTART=X
      GO TO 1
    9 IF (N.NE.6HXSTART) GO TO 10
      XSTART=X
      GO TO 1
   10 IF (N.NE.5HTSTOP) GO TO 11
      TSTOP=X
      GO TO 1
   11 IF (N.NE.4HSITE) GO TO 14
      DO 12 J=1,6
   12 NITL1(J)=NTEP(J)
      PRINT 65
      NTOTS=0
   13 READ 66, NTYPE(NTOTS+1),NSITE(NTOTS+1),XSITE(NTOTS+1),SITRAD(NTOTS
     1+1),PUP(NTOTS+1),TERFR(NTOTS+1)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 66, NTYPE(NTOTS+1),NSITE(NTOTS+1),XSITE(NTOTS+1),SITRAD(NTOT
     1S+1),PUP(NTOTS+1),TERFR(NTOTS+1)
      NTOTS=NTOTS+1
      GO TO 13
   14 IF (N.NE.3HSAM) GO TO 15
      J=X
      READ 67, NS(J),NSS(J),IR(J),HRAD(J),RTRK(J),RLOCK(J),ELMAX(J),ALTM
     1IN(J)
      PRINT 68
      PRINT 67, NS(J),NSS(J),IR(J),HRAD(J),RTRK(J),RLOCK(J),ELMAX(J),ALT
     1MIN(J)
      READ 64, TINIT(J),TISH(J),TINTER(J),TRELOAD(J),ASPMIN(J),ASPMAX(J)
     1,AZMAX(J),ECME(J)
      PRINT 69
      PRINT 64, TINIT(J),TISH(J),TINTER(J),TRELOAD(J),ASPMIN(J),ASPMAX(J
     1),AZMAX(J),ECME(J)
      PRINT 70
      READ 64, RADTRK(J),RADLOCK(J)
      PRINT 64, RADTRK(J),RADLOCK(J)
      GO TO 1
   15 IF (N.NE.3HRNG) GO TO 17
      K=0
      J=X
      RLETH(J)=-1.E+10
      ALTMAX(J)=-1.E+10
```

```
      PRINT 71
   16 READ 64, ELR(K+1,J),RNG(K+1,J)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 64, ELR(K+1,J),RNG(K+1,J)
      NRNG(J)=K=K+1
      RLETH(J)=AMAX1(RLETH(J),RNG(K,J)*COS(ELR(K,J)/57.2958))
      ALTMAX(J)=AMAX1(ALTMAX(J),RNG(K,J)*SIN(ELR(K,J)/57.2958))
      GO TO 16
   17 IF (N.NE.4HFUSE) GO TO 19
      K=0
      J=X
      PRINT 72
   18 READ 64, ELF(K+1,J),FUS(K+1,J)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 64, ELF(K+1,J),FUS(K+1,J)
      NFUS(J)=K=K+1
      GO TO 18
   19 CONTINUE
      IF (N.NE.2HDF) GO TO 22
      IF (X.LE.0.) GO TO 21
      DO 20 I=1,10
   20 DF(I)=X
      GO TO 1
   21 READ 73, DF
      PRINT 73, DF
      GO TO 1
   22 CONTINUE
      IF (N.NE.5HDEBUG) GO TO 23
      DEBUG=X
      GO TO 1
   23 CONTINUE
      IF (N.NE.6HDEBUG1) GO TO 24
      DEBUG1=X
      GO TO 1
   24 CONTINUE
      IF (N.NE.7HCORWDTH) GO TO 25
      CORWDTH=X
      GO TO 1
   25 IF (N.NE.4HPKSS) GO TO 26
      J=1
      IF (X.NE.0.) J=2
      READ 73, (PKSS(K,J),K=1,10)
      PRINT 73, (PKSS(K,J),K=1,10)
      GO TO 1
   26 IF (N.NE.4HXECM) GO TO 28
      J=0
      NXECM=0
   27 READ 64, XECM(J+1,1),XECM(J+1,2)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 64, XECM(J+1,1),XECM(J+1,2)
      NXECM=J=J+1
      GO TO 27
   28 CONTINUE
      IF (N.NE.7HCLUTTER) GO TO 29
```

```
      CLUTTER=X
      GO TO 1
   29 IF (N.NE.7HTERRAIN) GO TO 32
      NTER=0
   30 READ 64, TFRAC(NTER+1),ANTH(NTER+1)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 74
      PRINT 64, TFRAC(NTER+1),ANTH(NTER+1)
      NTER=NTER+1
      J=0
      NTERA(NTER)=0
      PRINT 75
   31 READ 64, TERRANE(J+1,1,NTER),TERRANE(J+1,2,NTER),TERRANE(J+1,3,NTE
     1R)
      IF (EOF(1).NE.0) GO TO 30
      PRINT 64, TERRANE(J+1,1,NTER),TERRANE(J+1,2,NTER),TERRANE(J+1,3,NT
     1ER)
      NTERA(NTER)=J=J+1
      GO TO 31
   32 CONTINUE
      IF (N.NE.4HAREA) GO TO 33
      AREA=X
      GO TO 1
   33 CONTINUE
      IF (N.NE.6HDEBUG2) GO TO 34
      DEBUG2=X
      GO TO 1
   34 CONTINUE
      IF (N.NE.7HTSTARTW) GO TO 35
      TSTART(2)=X
      GO TO 1
   35 CONTINUE
      IF (N.NE.6HTSTOPW) GO TO 36
      TSTOP(2)=X
      GO TO 1
   36 CONTINUE
      IF (N.NE.7HDXSTART) GO TO 37
      DXSTART=X
      GO TO 1
   37 IF (N.NE.7HNXSTART) GO TO 38
      NXSTART=X
      GO TO 1
   38 CONTINUE
      IF (N.NE.7HAVERAGE) GO TO 39
      AVERAGE=X
      GO TO 1
   39 CONTINUE
      IF (N.NE.4HDLEV) GO TO 41
      NDLEV=J=0
   40 READ 64, DLEV(J+1)
      IF (EOF(1).NE.0) GO TO 1
      PRINT 64, DLEV(J+1)
      NDLEV=J=J+1
      GO TO 40
```

```
41 CONTINUE
   IF (N.NE.7HSYMETRY) GO TO 42
   SYMETRY=X
   GO TO 1
42 CONTINUE
   IF (N.NE.6HEGRESS) GO TO 43
   READ 64, TEGRESS,FEGRESS
   PRINT 76
   PRINT 64, TEGRESS,FEGRESS
   GO TO 1
43 CONTINUE
   IF (N.NE.5HXYZTW) GO TO 48
   DO 44 J=1,6
44 NITL3(J)=NTEP(J)
   XSAMEW=X
   PRINT 63
   K=0
45 READ 64, (XYZ(K+1,J,2),J=1,3),T(K+1,2)
   IF (EOF(1).NE.0) GO TO 47
   PRINT 64, (XYZ(K+1,J,2),J=1,3),T(K+1,2)
   K=K+1
   IF (K.EQ.1) GO TO 45
   DO 46 J=1,3
46 DXYZDT(K,J,2)=(XYZ(K,J,2)-XYZ(K-1,J,2))/(T(K,2)-T(K-1,2))
   GO TO 45
47 NXYZT(2)=K
   GO TO 1
48 IF (N.NE.9HSIGNATURE) GO TO 51
   J=X
   NS=0
   PRINT 77
49 READ 64, SIGTH(NS+1,J),SIG(NS+1,J)
   IF (EOF(1).NE.0) GO TO 50
   PRINT 64, SIGTH(NS+1,J),SIG(NS+1,J)
   NS=NS+1
   GO TO 49
50 NSIG(J)=NS
   GO TO 1
51 IF (N.NE.TSTOPW) GO TO 52
   TSTOP(2)=X
   GO TO 1
52 CONTINUE
   IF (N.NE.6HMISLXT) GO TO 54
   PRINT 78
   J=X
   NXMISL(J)=K=0
53 READ 64, XMISL(K+1,J),TMISL(K+1,J)
   IF (EOF(1).NE.0) GO TO 1
   PRINT 64, XMISL(K+1,J),TMISL(K+1,J)
   NXMISL(J)=K=K+1
   GO TO 53
54 IF (N.NE.6HTSTOPP) GO TO 55
   TSTOPP=X
   GO TO 1
```

```
55 IF (N.NE.7HTSTARTP) GO TO 56
   TSTARTP=X
   GO TO 1
56 CONTINUE
   IF (N.NE.5HNOSIG) GO TO 57
   READ 79, NINTRPR
   PRINT 79, NINTRPR
   GO TO 1
57 CONTINUE
   IF (N.NE.7HFASTRUN) GO TO 58
   FASTRUN=X
   GO TO 1
58 CONTINUE
   IF (N.NE.TARGETY) GO TO 59
   TARGETY=X
   GO TO 1
59 CONTINUE
   PRINT 80
   IER=1
   GO TO 1
C
60 FORMAT (1H1)
61 FORMAT (A10,F10.0,6A10)
62 FORMAT (1X,A10,F10.4,10X,6A10)
63 FORMAT (9X,1HX,9X,1HY,9X,1HZ,9X,1HT)
64 FORMAT (8E10.3)
65 FORMAT (5X,5HNTYPE,5X,5HNSITE,5X,5HXSITE,5X,5HYSITE,7X,3HPUP,5X,5H
  1TERFR)
66 FORMAT (2I10,6F10.4)
67 FORMAT (3I10,5E10.3)
68 FORMAT (8X,2HNS,7X,3HNSS,8X,2HIR,6X,4HHRAD,6X,4HRTRK,5X,5HRLOCK,5X
  1,5HELMAX,4X,6HALTMIN)
69 FORMAT (5X,5HTINIT,6X,4HTISH,4X,6HTINTER,3X,7HTRELOAD,4X,6HASPMIN,
  14X,6HASPMAX,5X,5HAZMAX,6X,4HECME)
70 FORMAT (4X,6HRADTRK,3X,7HRADLOCK)
71 FORMAT (7X,3HELR,7X,3HRNG)
72 FORMAT (7X,3HELF,7X,3HFUS)
73 FORMAT (10F8.2)
74 FORMAT (5X,5HTFRAC,6X,4HANTH)
75 FORMAT (4X,6HAV ELF,4X,6HAV ELR,6X,4HPROB)
76 FORMAT (3X,7HTEGRESS,3X,7HFEGRESS)
77 FORMAT (5X,5HSIGTH,7X,3HSIG)
78 FORMAT (5X,5HXMISL,5X,5HTMISL)
79 FORMAT (10I8)
80 FORMAT (46H PREVIOUS CARD NOT IDENTIFIED, JOB TERMINATED )
   END
```

```
      SUBROUTINE INCOV (NSAM,TI,OFSET,ISHOT,TFIRE,TINT,TAQ,TS,L1)
C
C
C     THIS ROUTINE DETERMINES IF A SAM CAN SHOOT A MISSILE TO INTERCEPT
C     AT TIME TI
C
C
C     TI        TIME AT INTERCEPT
C     NSAM      INDEX NUMBER OF SAM TO BE CONSIDERED
C     OFSET     OFFSET DISTANCE
C     TFIRE     TIME SAM WOULD HAVE HAD TO FIRE FOR INTERCEPT AT TI
C     ISHOT     =1  CAN GET OFF A SHOT WITH INTERCEPT AT TI
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ISAM/ NTYPE(100),NSITE(100),XSITE(100),PUP(100),ELMIN(100,
     12),NTOTS,SITWDTH(100,3),AVSHOT(100,3),TERFR(100),SITRAD(100),ARSIT
     2(100)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      COMMON /D/ AZT,ELT,RNGT,ASPT,AZTF,ELTF,RNGTF,ASPTF,AZTA,ELTA,RNGTA
     1,ASPTA
      K=L1
      ISHOT=0
      NST=NTYPE(NSAM)
      IF (DF(NST).LE.0.) RETURN
C     FIND AZ,EL,ASP,RNG OF PENETRATOR WRT SITE AT TI
      NASP=1
      CALL COORD (NST,XSITE(NSAM),OFSET,TI,AZT,ELT,RNGT,ASPT,K)
      IF (RLETH(NST).LT.RNGT*COS(ELT/57.2985)) RETURN
      ALT=RNGT*SIN(ELT/57.2958)
      IF (ALT.LT.ALTMIN(NST)) RETURN
      NASP=0
C     CALCULATE FLYOUT TIME,FIRING TIME, ACQUISITION TIME
      TFIRE=TI-TRP(RNGT,XMISL(1,NST),TMISL(1,NST),NXMISL(NST))
      TAQ=TFIRE-TINT
      NN=10HTFIRE,TAQ
      IF (DEBUG.GT.0.) PRINT 12, NN,TFIRE,TAQ,TI,TS,TSTART
      IF (TFIRE.LT.TS) RETURN
      IF (TFIRE.LE.TSTARTP.O.TAQ.LT.TSTARTP) RETURN
C     CHECK IF WITHIN LETHAL ENVELOPE
      NN=10H ENVEL CK
      IF (DEBUG.GT.0.) PRINT 12, NN,TI,OFSET,AZT,ELT,RNGT,ASPT,RMIN,RMAX
     1,FLOAT(K)
      IF (AZT.GT.AZMAX(NST)) RETURN
```

```
      L=1
      IF (AZT.GT.90.) L=2
      IF (IR(NST).LE.0.A.(ELT.LT.ELMIN(NSAM,L).O.ELT.GT.ELMAX(NST))) RET
     1URN
      RMAX=TRP(ELT,ELR(1,NST),RNG(1,NST),NRNG(NST))
      IF (RNGT.GT.RMAX) RETURN
      RMIN=TRP(ELT,ELF(1,NST),FUS(1,NST),NFUS(NST))
      IF (RNGT.LT.RMIN) RETURN
C     FIND AZ,EL,RNG,ASP OF PENETRATOR WRT SITE AT TFIRE
      K=L1
      IF (NSYSOP.EQ.3.A.TFIRE.LT.TSTART(2).A.K.EQ.2) K=1
      K1=K
      IF (IR(NST).GT.0) K1=K1+2
      CALL COORD (NST,XSITE(NSAM),OFSET,TFIRE,AZTF,ELTF,RNGTF,ASPTF,K)
      IF (NINTRPR(NST).GT.0) GO TO 1
      APEN=TRP(ASPTF,SIGTH(1,K1),SIG(1,K1),NSIG(K1))
      RMTF=SQRT(APEN/RLOCK(NST))
      IF(IR(NST).LE.0)RMTF=SQRT(RMTF)
      IF (NINTRPR(NST).LT.0) GO TO 2
      GO TO 3
    1 RMTF=RADLOCK(NST)
    2 IF (ASPTF.LT.ASPMIN(NST).O.ASPTF.GT.ASPMAX(NST)) RETURN
    3 CONTINUE
      NN=10HFIRE CK
      IF (DEBUG.GT.0.) PRINT 12, NN,AZTF,ELTF,RNGTF,ASPTF,FLOAT(K),FLOAT
     1(K1),RNTF
C     CHECK IF WITHIN FIRING CONSTRAINTS AT TFIRE
      L=1
      IF (AZTF.GT.90.) L=2
      IF (RNGTF.GT.RMTF.O.ELTF.LT.ELMIN(NSAM,L).O.ELTF.GT.ELMAX(NST)) RE
     1TURN
C     CHECK IF OBSERVED AT TAQ
      K=L1
      IF (NSYSOP.EQ.3.A.TAQ.LT.TSTART(2).A.K.EQ.2) K=1
      K1=K
      IF (IR(NST).GT.0) K1=K1+2
      CALL COORD (NST,XSITE(NSAM),OFSET,TAQ,AZTA,ELTA,RNGTA,ASPTA,K)
      IF (NINTRPR(NST).GT.0) GO TO 4
      APEN=TRP(ASPTA,SIGTH(1,K1),SIG(1,K1),NSIG(K1))
      RNTA=SQRT(APEN/RTRK(NST))
      IF(IR(NST).LE.0)RMTA=SQRT(RMTA)
      GO TO 5
    4 RMTA=RADTRK(NST)
    5 CONTINUE
      NN=10HOBS CK
      IF (DEBUG.GT.0.) PRINT 12, NN,AZTA,ELTA,RNGTA,ASPTA,FLOAT(K),FLOAT
     1(K1),RMTA
      L=1
      IF (AZTA.GT.90.) L=2
      IF (RNGTA.GT.RMTA.O.ELTA.LT.ELMIN(NSAM,L).O.ELTA.GT.ELMAX(NST)) RE
     1TURN
      IF (IR(NST).GT.0.) GO TO 11
C     CHECK IF PENETRATOR IS IN RADAR COVERAGE DURRING FLIGHT TIME
      DELT=.25*(TI-TFIRE)
```

```
      NIN=0
      DO 10 I=1,3
      TCK=TFIRE+I*DELT
      K=L1
      IF (NSYSOP.EQ.3.A.TCK.LT.TSTART(2).A.K.EQ.2) K=1
      K1=K
      IF (IR(NST).GT.0) K1=K1+2
      CALL COORD (NST,XSITE(NSAM),OFSET,TCK,AZC,ELC,RNGC,ASPC,K)
      IF (NINTRPR(NST).GT.0) GO TO 6
      APEN=TRP(ASPC ,SIGTH(1,K1),SIG(1,K1),NSIG(K1))
      RMTC=(APEN/RTRK(NST))**(.25)
      IF (NINTRPR(NST).LT.0) GO TO 7
      GO TO 8
    6 RMTC=RADTRK(NST)
    7 IF (ASPC.GT.ASPMAX(NST).O.ASPC.LT.ASPMIN(NST)) GO TO 9
    8 CONTINUE
      L=1
      IF (AZC.GT.90.) L=2
      IF (RNGC.LE.RMTC.A.ELC.GE.ELMIN(NSAM,L).A.ELC.LE.ELMAX(NST)) GO TO
     1 10
    9 CONTINUE
      NIN=NIN+1
   10 CONTINUE
      IF (NIN.GE.2) RETURN
   11 CONTINUE
C     SAM CAN GET OFF A SHOT
      ISHOT=1
      NN=10HGOOD SHOT
      IF (DEBUG.GT.0.) PRINT 12, NN
      RETURN
C
   12 FORMAT (1X,A10,12F10.4)
      END
```

```
      SUBROUTINE COORD (N,XSITE,OFSET,TC,AZ,EL,RNGE,ASP,K)
C
C
C     THIS ROUTINE MAPS TRAJECTORIES INTO 3 DIMENSIONAL SPHERICAL EARTH
C     COORDINATE SYSTEM WITH SAM AT (0,0,8495)KM
C     CALCULATES AZ,EL,RNG,ASP OF PENETRATOR WRT SITE FOR SPHEERICAL EAR
C     TH
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      COMMON /INTER/ L
      DIMENSION US(3), UP(3), USP(3), XYZTC(3), UP1(3), VEL(3)
C     COORDINATE SYSTEM ORIGIN AT CENTER OF SPHERE
C     X  ALONG CORRODOR
C     Z  VERTICAL THROUGH SAM SITE
C     Y  ORTHOGONAL
      ASP=0.
      ER=6378.165
      RTD=57.29578
      ANG=180.
      IF (AREA.LT.0.) ANG=OFSETD
      UP1(1)=COS(ANG/RTD)
      UP1(2)=SIN(ANG/RTD)
      USM=US(3)=ER+HRAD(N)
      US(1)=US(2)=0.
      CALL TRP1 (TC,XYZTC,K)
      XYZTCM=ER+XYZTC(3)
      SX=XYZTC(1)-XSITE+XSTART
      UP(1)=XYZTCM*SIN(SX/XYZTCM)
      SY=XYZTC(2)+OFSET
      UP(2)=XYZTCM*SIN(SY/XYZTCM)
      UP(3)=SQRT(XYZTCM**2-UP(1)**2-UP(2)**2)
      RNGE=0.
      DO 1 I=1,3
      USP(I)=UP(I)-US(I)
    1 RNGE=RNGE+USP(I)**2
      RNGE=SQRT(RNGE)
      EL=ASIN(USP(3)/RNGE)*RTD
      AZ=0.
      AZLEN=SQRT(USP(1)**2+USP(2)**2)
      IF (AZLEN.LE.0.) GO TO 2
```

```
      DOT=(UP1(1)*USP(1)+UP1(2)*USP(2))/AZLEN
      AZ=ACOS(DOT)*RTD
      AZ=AMOD(AZ+720.,360.)
    2 CONTINUE
      ASP=0.
      IF (NASP.GT.0) RETURN
      IF (ASPMAX(N)-ASPMIN(N).GE.180..A.NINTRPR(N).GT.0) RETURN
      IF (TC.GT.T(L-1,K)) GO TO 4
      SIGN=1.
      DO 3 I=1,3
    3 UP1(I)=XYZ(L,I,K)
      GO TO 6
    4 SIGN=-1.
      DO 5 I=1,3
    5 UP1(I)=XYZ(L-1,I,K)
    6 CONTINUE
      UP1M=ER+UP1(3)
      SX=UP1(1)-XSITE+XSTART
      UP1(1)=UP1M*SIN(SX/UP1M)
      SY=UP1(2)+OFSET
      UP1(2)=UP1M*SIN(SY/UP1M)
      UP1(3)=SQRT(UP1M**2-UP1(1)**2-UP1(2)**2)
      DOT=VELM=0.
      DO 7 I=1,3
      VEL(I)=SIGN*(UP1(I)-UP(I))
      VELM=VELM+VEL(I)**2
    7 DOT=DOT+VEL(I)*USP(I)
      VELM=SQRT(VELM)
      ASP=180.-ACOS(DOT/(VELM*RNGE))*RTD
      RETURN
      END
```

```
      FUNCTION TRP (X1,X,Y,N)
      COMMON /INTER/ I1
      DIMENSION X(N), Y(N)
      IF (N.LE.2) GO TO 2
      DO 1 I=2,N
      IF (X1.LE.X(I)) GO TO 3
    1 CONTINUE
    2 I=N
    3 CONTINUE
      TRP=Y(I-1)+(X1-X(I-1))*(Y(I)-Y(I-1))/(X(I)-X(I-1))
      I1=I
      RETURN
      END
```

```
      SUBROUTINE TRP1 (X1,Z,K)
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /INTER/ I
      DIMENSION Z(3)
      N=NXYZT(K)
      L=IS(K)
      IF (X1.LT.T(L-1,K)) GO TO 2
      DO 1 I=L,N
      IF (X1.LE.T(I,K)) GO TO 4
    1 CONTINUE
      I=N
      GO TO 4
    2 DO 3 M=2,L
      I=L-M+2
      IF (X1.GE.T(I-1,K)) GO TO 4
    3 CONTINUE
      I=2
    4 CONTINUE
      DX=X1-T(I-1,K)
      DO 5 J=1,3
    5 Z(J)=XYZ(I-1,J,K)+DX*DXYZDT(I,J,K)
      IS(K)=I
      RETURN
      END
```

```
      FUNCTION SGN (X)
      IF (X) 1,2,3
    1 SGN=-1.
      RETURN
    2 SGN=0.
      RETURN
    3 SGN=1.
      RETURN
      END
```

```
      SUBROUTINE TIMEIN (I,DT,N)
C
C
C     THIS ROUTINE DETERMINES ROUGH ESTIMATES OF THE INTERVALS THAT A
C     TRAJECTORY IS IN RANGE OF A SAM SITE
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      COMMON /ISAM/ NTYPE(100),NSITE(100),XSITE(100),PUP(100),ELMIN(100,
     12),NTOTS,SITWDTH(100,3),AVSHOT(100,3),TERFR(100),SITRAD(100),ARSIT
     2(100)
      COMMON /ASAM/ HRAD(10),RTRK(10),ELMAX(10),TINIT(10),TINTER(10),NS(
     110),NSS(10),TRELOAD(10),AVVEL(10),ASPMIN(10),ASPMAX(10),AZMAX(10),
     2RNG(20,10),ELR(20,10),NRNG(10),FUS(20,10),ELF(20,10),NFUS(10),IR(1
     30),TISH(10),ECME(10),ALTMIN(10),ALTMAX(10),SIGTH(20,4),SIG(20,4),N
     4SIG(4),RLOCK(10),XMISL(20,10),TMISL(20,10),NXMISL(10),RADTRK(10),R
     5ADLOCK(10)
      COMMON /PARM/ DF(10),DEBUG,FLTWTH(2,2),RLETH(10),DEBUG1,TITLE(6),P
     1KSS(10,2),CORWDTH,CLUTTER,TERRANE(10,3,10),NTERA(10),NTER,TFRAC(10
     2),ANTH(10),AREA,DEBUG2,OFSETD,RELEASE,VELPEN,DLEV(10),NDLEV,AVERAG
     3E,NITL1(6),NITL2(6),SYMETRY,TEGRESS,FEGRESS,NITL3(6),NINTRPR(10),N
     4ASP,FASTRUN
      DIMENSION VECI(2), THMM(2)
      DIMENSION IN(2), SVX(2), SVY(2)
      DIMENSION XYZTO(3)
      EQUIVALENCE (XYZTO(1),XTO), (XYZTO(2),YTO), (XYZTO(3),ZTO)
      DPR=57.2957
      XMIN=XSITE(I)-RLETH(N)-XSTART
      IF (AREA.LE.0.) XMIN=SITRAD(I)-RLETH(N)
      XMAX=XMIN+2.*RLETH(N)
      ARSIT(I)=180.
      IF (XMIN.GE.0..A.AREA.LE.0.) ARSIT(I)=ASIN(RLETH(N)/SITRAD(I))*DPR
      NTIN=0
      FLTWTH(1,1)=-1.E+10
      FLTWTH(1,2)=-1.E+10
      FLTWTH(2,2)=1.E+10
      FLTWTH(2,1)=1.E+10
      J=1
      TO=TSTARTP-DT
      K=0
      INL=0
    1 K=K+1
      IF (K.GT.2) K=1
      IF (K.EQ.1) TO=TO+DT
      IF (K.EQ.1.A.NSYSOP.EQ.2) GO TO 5
      IF (K.EQ.2.A.NSYSOP.EQ.1) GO TO 5
      IF (NSYSOP.NE.2.A.K.EQ.2.A.(TO.LT.TSTART(2).O.TO.GT.TSTOP(2))) GO
     1TO 5
      IF (TO.GT.TSTOPP+.5*DT) GO TO 9
      IN(K)=0
      CALL TRP1 (TO,XYZTO,K)
      IF (AREA.GT.0.) GO TO 3
```

```
      IF (TO.EQ.TSTART) GO TO 2
      SVY(K)=YTO
      SVX(K)=XTOT
    2 CONTINUE
      XTOT=XTO+XSTART
      XTO=SQRT((XTO+XSTART)**2+YTO**2)
      IF (TO.GT.TSTART) GO TO 3
      SVX(K)=XTOT
      SVY(K)=YTO
    3 CONTINUE
      IF (XTO.GE.XMIN.A.XTO.LE.XMAX.A.ZTO.GE.ALTMIN(N).A.ZTO.LE.ALTMAX(N
     1)) IN(K)=1
      IF (IN(K).GT.0) FLTWTH(1,K)=AMAX1(FLTWTH(1,K),YTO)
      IF (IN(K).GT.0) FLTWTH(2,K)=AMIN1(FLTWTH(2,K),YTO)
      IF (AREA.GT.0.O.IN(K).EQ.0.O.INL.GT.0) GO TO 4
      VMAG=SQRT(SVX(K)**2+SVY(K)**2)
      VECI(1)=SVX(K)/VMAG
      VECI(2)=SVY(K)/VMAG
      THETAI=ATAN2(VECI(2),VECI(1))*DPR
      THMM(1)=THMM(2)=THETAI
      INL=1
      GO TO 6
    4 IF (INL.LE.0.O.AREA.GT.0.) GO TO 6
      VM=XTOT**2+YTO**2
      IF (VM.LE.0.) GO TO 6
      VX=XTOT/VM
      VY=YTO/VM
      DOT=-VECI(2)*VX+VECI(1)*VY
      DANG=90.-ACOS(DOT)*DPR
      THETAI=THETAI+DANG
      THMM(1)=AMIN1(THMM(1),THETAI)
      THMM(2)=AMAX1(THMM(2),THETAI)
      VECI(1)=VX
      VECI(2)=VY
      GO TO 6
    5 IN(K)=0
    6 CONTINUE
      IF (K.EQ.1) GO TO 1
      GO TO (7,8), J
    7 IF (IN(1).EQ.0.A.IN(2).EQ.0) GO TO 1
      J=2
      NTIN=NTIN+1
      TIN(NTIN,1)=AMAX1(TO-DT,TSTARTP)
      INL=1
      GO TO 1
    8 IF (IN(1).GT.0.O.IN(2).GT.0) GO TO 1
      J=1
      TIN(NTIN,2)=TO
      INL=0
      THIN(NTIN,1)=THMM(1)-ARSIT(I)
      THIN(NTIN,2)=THMM(2)+ARSIT(I)
      GO TO 1
    9 CONTINUE
      IF (NTIN.LE.0.O.INL.LE.0) GO TO 10
```

```
      TIN(NTIN,2)=TSTOPP
      THIN(NTIN,1)=THMM(1)-ARSIT(I)
      THIN(NTIN,2)=THMM(2)+ARSIT(I)
   10 CONTINUE
      IF (NTIN.LE.0.O.AREA.GT.0.) GO TO 12
      DO 11 J=1,NTIN
      IF (XMIN.GE.0..A.THIN(J,2)-THIN(J,1).LT.360.) GO TO 11
      THIN(J,1)=0.
      THIN(J,2)=360.
   11 CONTINUE
      IF (AREA.LE.0..A.NTIN.GT.1) CALL COLAPS
   12 CONTINUE
      IF (DEBUG1.GT.0..O.DEBUG2.GT.0.) PRINT 13, NTIN,(TIN(K,1),TIN(K,2)
     1,THIN(K,1),THIN(K,2),K=1,NTIN)
      RETURN
C
   13 FORMAT (8H TIN/OUT,I10,(4F10.2))
      END
```

```
      SUBROUTINE COLAPS
C
C
C     THIS ROUTINE COMBINES OVERLAPPING ANGULAR COVERAGE INTERVALS
C
C
      COMMON /TRAJ/ NXYZT(2),XYZ(1000,3,2),T(1000,2),XSTART,TSTART(2),TS
     1TOP(2),TIN(10,2),NTIN,XECM(20,2),NXECM,THIN(10,2),DXSTART,NXSTART,
     2XSAMEP,XSAMEW,NSYSOP,TSTOPP,TSTARTP,TARGETY,DXYZDT(1000,3,2),IS(2)
      N1=NTIN-1
      N=NTIN
      DO 8 I=1,N1
      IP=I+1
      DO 7 J=2,N
      K=N+IP-J
      TI1=AMOD(THIN(I,1)+720.,360.)
      TI2=AMOD(THIN(I,2)+720.,360.)
      TK1=AMOD(THIN(K,1)+720.,360.)
      TK2=AMOD(THIN(K,2)+720.,360.)
      TIM=(TI2+TI1)*.5
      TID=(TI2-TI1)*.5
      IF (TID.GE.0.) GO TO 1
      TID=180.+TID
      TIM=AMOD(TIM+900.,360.)
    1 TKM=(TK2+TK1)*.5
      TKD=(TK2-TK1)*.5
      IF (TKD.GE.0.) GO TO 2
      TKD=180.+TKD
      TKM=AMOD(TKM+900.,360.)
    2 CONTINUE
      AM=AMAX1(TKM,TIM)-AMIN1(TKM,TIM)
      IF (AM.GT.180.) AM=360.-AM
      IF (AM.GT.TKD+TID) GO TO 7
      TIN(I,1)=AMIN1(TIN(I,1),TIN(K,1))
      TIN(I,2)=AMAX1(TIN(I,2),TIN(K,2))
      IF (TKM.GE.TIM) GO TO 3
      S=TKM
      TKM=TIM
      TIM=S
      S=TKD
      TKD=TID
      TID=S
    3 CONTINUE
      IF (TKM-TIM.GT.180.) TIM=TIM+360.
      THIN(I,1)=AMIN1(TKM-TKD,TIM-TID)
      THIN(I,2)=AMAX1(TKM+TKD,TIM+TID)
      IF (THIN(I,2)-THIN(I,1).LT.360.) GO TO 4
      THIN(I,1)=0.
      THIN(I,2)=360.
    4 CONTINUE
      N=N-1
      N1=N1-1
      IF (K.EQ.N+1) GO TO 6
      L1=K+1
```

```
      L2=N+1
      DO 5 L=L1,L2
      THIN(L-1,1)=THIN(L,1)
      THIN(L-1,2)=THIN(L,2)
    5 CONTINUE
    6 CONTINUE
    7 CONTINUE
    8 CONTINUE
      NTIN=N
      RETURN
      END
```

APPENDIX B

SAMPLE PROBLEM FOR SURVIVE

A. INPUT CARD LISTING

CARD IMAGES FOR SAMPLE PROBLEM

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

TITLE               SAMPLE PROBLEM FOR SURVIVE MODEL
DLEV                DEFENCE LEVEL EDITS
.25
.5
.75
1.
(EOR CARD)
SAM        1.        DEFINE PARAMETERS FOR TYPE 1 SAM
         3         2         1 .005                        90.       .03
8.        1.        10.       500.      45.       181.        181.      .1
30.       25.
MISLXT     1.       MISSILE TIME TO DISTANCE PROFILE
0.        0.
100.      100.
(EOR CARD)
RNG       1.        LETHAL ENVELOPE
0.        10.
45.       8.
70.       7.
90.       6.
(EOR CARD)
FUSE      1.        DEAD ZONE
0.        .3
90.       .3
(EOR CARD)
CLUTTER   .1        GROUND CLUTTER ANGLE
TERRAIN             DEFINE TERRAIN MASK ANGLE DISTRIBUTIONS
50.       .005
.15       2.5       .5
.25       2.5       .5
(EOR CARD)
(EOR CARD)
XECM                ECM INTERVALS
0.         20.
50.        100.
(EOR CARD)
PKSS                SINGLE SHOT KILL PROBABILITY FOR PENETRATOR
```

CARD IMAGES FOR SAMPLE PROBLEM

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890


.2      0.      0.      0.      0.      0.      0.      0.      0.      0.
PKSS      1.        SINGLE SHOT KILL PROBABILITY FOR WEAPON
.1      0.      0.      0.      0.      0.      0.      0.      0.      0.
AREA      1.        SELECT RECTANGULAR COORDINATES
AVERAGE       1.    SELECT EXPECTED VALUE METHOD
CORWDTH   20.       CORRIDOR WIDTH
FASTRUN    3.       TIME STEP FACTOR
DF                  INDIVIDUAL DEFENSE LEVEL
1.      0.      0.      0.      0.      0.      0.      0.      0.      0.
SYMETRY        1.   SELECT HOMOGENEOUS BOUNDARY CONDITION
NOSIG               SELECT MAX RANGE CRITERIA FOR SENSOR
       1       0       0       0       0       0       0       0       0       0
XYZT      -.1       AIRCRAFT INGRESS, 180 DEGREE TURN, EGRESS
 -500.0000    0.0000     .0750-1800.0000
    0.0000    0.0000     .0750    0.0000
     .4558     .0399     .0750    1.6481
     .8978     .1583     .0750    3.2962
    1.3126     .3517     .0750    4.9443
    1.6874     .6142     .0750    6.5924
    2.0110     .9377     .0750    8.2405
    2.2734    1.3126     .0750    9.8886
    2.4668    1.7273     .0750   11.5367
    2.5853    2.1693     .0750   13.1848
    2.6251    2.6251     .0750   14.8329
    2.5853    3.0810     .0750   16.4810
    2.4668    3.5230     .0750   18.1291
    2.2734    3.9377     .0750   19.7772
    2.0110    4.3125     .0750   21.4253
    1.6874    4.6361     .0750   23.0734
    1.3126    4.8986     .0750   24.7215
     .8978    5.0920     .0750   26.3697
     .4558    5.2104     .0750   28.0178
    0.0000    5.2503     .0750   29.6659
 -500.0000    5.2503     .0750 1829.6659
(EOR CARD)
EGRESS              DEFINE EGRESS TIME AND SHOT FRACTION
30.       .5
```

                              CARD IMAGES FOR SAMPLE PROBLEM

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

TSTART    -1800.    AIRCRAFT FLIGHT FLIGHT START TIME
TSTOP     1829.     AIRCRAFT FLIGHT STOP TIME
XYZTW       0.       WEAPON TRAJECTORY - CIRCULAR
0.        0.        .075      0.
1.414     .585      .128      5.
2.        2.        .181      10.
1.414     3.414     .234      15.
0.        4.        .288      20.
-1.414    3.414     .341      25.
-2.       2.        .394      30.
-1.414    .585      .447      35.
0.        0.        .5        40.
(EOR CARD)
TSTARTW             WEAPON FLIGHT START TIME
TSTOPW    40.       WEAPON FLIGHT STOP TIME
SITE                DEFINE SITE LOCATIONS
         1         1  5.       0.        .5       50.
         1         1 10.       0.        .5       50.
         1         1 25.       0.        .5       50.
         1         1 35.       0.        .5       50.
         1         1 45.       0.        .5       50.
         1         1 60.       0.        .5       50.
         1         1 70.       0.        .5       50.
         1         1 80.       0.        .5       50.
(EOR CARD)
NXSTART     4.      NUMBER OF TARGET LOCATIONS
DXSTART     10.     TARGET LOCATION SPACING
XSTART    10.       FIRST TARGET LOCATION
ENDCASE
ENDJOB
```

B. OUTPUT LISTING

```
TITLE           0.0000          SAMPLE PROBLEM FOR SURVIVE MODEL
DLEV            0.0000          DEFENCE LEVEL EDITS
 .250E+00
 .500E+00
 .750E+00
 .100E+01
SAM             1.0000           DEFINE PARAMETERS FOR TYPE 1 SAM
       NS          NSS         IR       HRAD       RTRK      RLOCK      ELMAX     ALTMIN
        3            2          1  .500E-02 0.          0.          .900E+02  .300E-01
    TINIT         TISH     TINTER    TRELOAD     ASPMIN     ASPMAX      AZMAX       ECME
 .800E+01  .100E+01  .100E+02  .500E+03  .450E+02  .181E+03  .181E+03  .100E+00
   RADTRK    RADLOCK
 .300E+02  .250E+02
MISLXT          1.0000          MISSILE TIME TO DISTANCE PROFILE
    XMISL        TMISL
0.         0.
 .100E+03  .100E+03
RNG             1.0000          LETHAL ENVELOPE
      ELR          RNG
0.          .100E+02
 .450E+02  .800E+01
 .700E+02  .700E+01
 .900E+02  .600E+01
FUSE            1.0000          DEAD ZONE
      ELF          FUS
0.          .300E+00
 .900E+02  .300E+00
CLUTTER          .1000          GROUND CLUTTER ANGLE
TERRAIN         0.0000          DEFINE TERRAIN MASK ANGLE DISTRIBUTIONS
    TFRAC         ANTH
 .500E+02  .500E-02
   AV ELF       AV ELR       PROB
 .150E+00  .250E+01  .500E+00
 .250E+00  .250E+01  .500E+00
XECM            0.0000          ECM INTERVALS
0.          .200E+02
 .500E+02  .100E+03
PKSS            0.0000          SINGLE SHOT KILL PROBABILITY FOR PENETRATOR
      .20     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
PKSS            1.0000          SINGLE SHOT KILL PROBABILITY FOR WEAPON
      .10     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
AREA            1.0000          SELECT RECTANGULAR COORDINATES
AVERAGE         1.0000          SELECT EXPECTED VALUE METHOD
CORWDTH        20.0000          CORRIDOR WIDTH
FASTRUN         3.0000          TIME STEP FACTOR
DF              0.0000          INDIVIDUAL DEFENSE LEVEL
     1.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
SYMETRY         1.0000          SELECT HOMOGENEOUS BOUNDARY CONDITION
NOSIG           0.0000          SELECT MAX RANGE CRITERIA FOR SENSOR
        1        0        0        0        0        0        0        0        0        0
XYZT            -.1000          AIRCRAFT INGRESS, 180 DEGREE TURN, EGRESS
        X            Y          Z          T
-.500E+03 0.          .750E-01 -.180E+04
0.         0.          .750E-01 0.
 .456E+00  .399E-01  .750E-01  .165E+01
 .898E+00  .158E+00  .750E-01  .330E+01
 .131E+01  .352E+00  .750E-01  .494E+01
 .169E+01  .614E+00  .750E-01  .659E+01
 .201E+01  .938E+00  .750E-01  .824E+01
 .227E+01  .131E+01  .750E-01  .989E+01
 .247E+01  .173E+01  .750E-01  .115E+02
 .259E+01  .217E+01  .750E-01  .132E+02
```

```
 .263E+01  .263E+01  .750E-01  .148E+02
 .259E+01  .308E+01  .750E-01  .165E+02
 .247E+01  .352E+01  .750E-01  .181E+02
 .227E+01  .394E+01  .750E-01  .198E+02
 .201E+01  .431E+01  .750E-01  .214E+02
 .169E+01  .464E+01  .750E-01  .231E+02
 .131E+01  .490E+01  .750E-01  .247E+02
 .898E+00  .509E+01  .750E-01  .264E+02
 .456E+00  .521E+01  .750E-01  .280E+02
0.         .525E+01  .750E-01  .297E+02
-.500E+03  .525E+01  .750E-01  .183E+04
EGRESS       0.0000             DEFINE EGRESS TIME AND SHOT FRACTION
  TEGRESS   FEGRESS
 .300E+02  .500E+00
TSTART    -1800.0000            AIRCRAFT FLIGHT FLIGHT START TIME
TSTOP      1829.0000            AIRCRAFT FLIGHT STOP TIME
XYZTW        0.0000              WEAPON TRAJECTORY - CIRCULAR
       X          Y          Z          T
0.         0.        .750E-01  0.
 .141E+01  .585E+00  .128E+00  .500E+01
 .200E+01  .200E+01  .181E+00  .100E+02
 .141E+01  .341E+01  .234E+00  .150E+02
0.         .400E+01  .288E+00  .200E+02
-.141E+01  .341E+01  .341E+00  .250E+02
-.200E+01  .200E+01  .394E+00  .300E+02
-.141E+01  .585E+00  .447E+00  .350E+02
0.         0.        .500E+00  .400E+02
TSTARTW      0.0000             WEAPON FLIGHT START TIME
TSTOPW      40.0000             WEAPON FLIGHT STOP TIME
SITE         0.0000             DEFINE SITE LOCATIONS
   NTYPE      NSITE      XSITE      YSITE        PUP      TERFR
       1          1     5.0000     0.0000      .5000    50.0000
       1          1    10.0000     0.0000      .5000    50.0000
       1          1    25.0000     0.0000      .5000    50.0000
       1          1    35.0000     0.0000      .5000    50.0000
       1          1    45.0000     0.0000      .5000    50.0000
       1          1    60.0000     0.0000      .5000    50.0000
       1          1    70.0000     0.0000      .5000    50.0000
       1          1    80.0000     0.0000      .5000    50.0000
NXSTART      4.0000             NUMBER OF TARGFT LOCATIONS
DXSTART     10.0000             TARGET LOCATION SPACING
XSTART      10.0000             FIRST TARGET LOCATION
ENDCASE      0.0000

------------------------------------------------------------------------------------------------------------------------------
RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE
XSTART =      10.00
                         SURVIVAL PROBABILITY WITHOUT ECM                              SURVIVAL PROBABILITY WITH ECM
 DEFENCE LEVEL A/C TO RELEASE    A/C COMPLETE      WEAPON/REL A/C TO RELEASE    A/C COMPLETE     WEAPON/REL   A/C+W W/O ECM
          .250           .901            .856            .964           .990            .985           .996            .954
          .500           .808            .723            .929           .980            .970           .993            .910
          .750           .719            .606            .894           .970            .956           .989            .867
         1.000           .638            .513            .860           .960            .941           .986            .825
SITE    X OR TH     Y OR R LETH WDTH  AV SHOTS LETH WDTH  AV SHOTS LETH WDTH  AV SHOTS
                           A/C TO DELIVERY         A/C COMPLETE     WEAPON (REL TO TAR)
   1      5.000      0.000     19.000      2.421     23.000      2.674     12.000      1.833
   1     10.000      0.000     18.000      1.944     23.000      2.533     13.000      2.769
   1     25.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
   1     35.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
   1     45.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
   1     60.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
   1     70.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
   1     80.000      0.000      0.000      0.000      0.000      0.000      0.000      0.000
```

RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE
XSTART = 20.00

DEFENCE LEVEL	Without ECM: A/C TO RELEASE	Without ECM: A/C COMPLETE	Without ECM: WEAPON/REL	With ECM: A/C TO RELEASE	With ECM: A/C COMPLETE	With ECM: WEAPON/REL	A/C+W W/O ECM
.250	.868	.791	.968	.985	.939	.968	.954
.500	.746	.612	.937	.971	.879	.937	.909
.750	.634	.474	.905	.956	.821	.905	.866
1.000	.545	.368	.875	.942	.766	.875	.825

SITE	X OR TH	Y OR R	LETH WDTH (A/C TO DELIVERY)	AV SHOTS (A/C TO DELIVERY)	LETH WDTH (A/C COMPLETE)	AV SHOTS (A/C COMPLETE)	LETH WDTH (WEAPON (REL TO TAR))	AV SHOTS (WEAPON (REL TO TAR))
1	5.000	0.000	19.000	2.842	23.000	3.130	0.000	0.000
1	10.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	25.000	0.000	1.000	1.000	16.500	2.139	19.000	2.664
1	35.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	45.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	60.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	70.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	80.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE
XSTART = 30.00

DEFENCE LEVEL	Without ECM: A/C TO RELEASE	Without ECM: A/C COMPLETE	Without ECM: WEAPON/REL	With ECM: A/C TO RELEASE	With ECM: A/C COMPLETE	With ECM: WEAPON/REL	A/C+W W/O ECM
.250	.819	.730	.955	.929	.866	.955	.887
.500	.661	.517	.911	.859	.744	.911	.783
.750	.525	.366	.868	.791	.635	.868	.687
1.000	.421	.260	.827	.729	.543	.827	.603

SITE	X OR TH	Y OR R	LETH WDTH (A/C TO DELIVERY)	AV SHOTS (A/C TO DELIVERY)	LETH WDTH (A/C COMPLETE)	AV SHOTS (A/C COMPLETE)	LETH WDTH (WEAPON (REL TO TAR))	AV SHOTS (WEAPON (REL TO TAR))
1	5.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	10.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	25.000	0.000	19.000	2.421	23.000	2.674	12.000	1.833
1	35.000	0.000	1.000	1.000	16.500	2.139	19.000	2.664
1	45.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	60.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	70.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	80.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

RESULTS AVERAGED OVER ALL OFFSETS APPLICABLE
XSTART = 40.00

DEFENCE LEVEL	Without ECM: A/C TO RELEASE	Without ECM: A/C COMPLETE	Without ECM: WEAPON/REL	With ECM: A/C TO RELEASE	With ECM: A/C COMPLETE	With ECM: WEAPON/REL	A/C+W W/O ECM
.250	.763	.664	.955	.866	.788	.955	.827
.500	.571	.423	.911	.743	.609	.911	.677
.750	.419	.270	.868	.631	.469	.868	.548
1.000	.312	.174	.827	.539	.362	.827	.446

SITE	X OR TH	Y OR R	LETH WDTH (A/C TO DELIVERY)	AV SHOTS (A/C TO DELIVERY)	LETH WDTH (A/C COMPLETE)	AV SHOTS (A/C COMPLETE)	LETH WDTH (WEAPON (REL TO TAR))	AV SHOTS (WEAPON (REL TO TAR))
1	5.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	10.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	25.000	0.000	19.000	2.842	23.000	3.152	0.000	0.000
1	35.000	0.000	19.000	2.421	23.000	2.674	12.000	1.833
1	45.000	0.000	1.000	1.000	16.500	2.139	19.000	2.664
1	60.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	70.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	80.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

```
****************************************************************************************************
AVERAGES FOR PREVIOUS    4 CASES  FIRST XSTART =      10.00
SAMPLE PROBLEM FOR SURVIVE MODEL
DEFINE SITE LOCATIONS
AIRCRAFT INGRESS, 180 DEGREE TURN, EGRESS
 WEAPON TRAJECTORY - CIRCULAR
AREA=  1.0 CORRIDOR WIDTH      20.00 XSTART=      10.00
START TIMES  -1800.00       0.00  STOP TIMES   1829.00      40.00
```

DEFENCE LEVEL	SURVIVAL PROBABILITY WITHOUT ECM A/C TO RELEASE	A/C COMPLETE	WEAPON/REL	SURVIVAL PROBABILITY WITH ECM A/C TO RELEASE	A/C COMPLETE	WEAPON/REL	A/C+W W/O ECM
.250	.838	.760	.961	.942	.895	.969	.905
.500	.696	.569	.922	.888	.801	.938	.819
.750	.574	.429	.884	.837	.720	.907	.740
1.000	.479	.329	.847	.793	.653	.879	.672